

11.07.03

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 3月14日

出 願 番 号

Application Number:

特願2003-069375

[ST.10/C]:

[JP2003-069375]

出 願 人

Applicant(s):

株式会社産学連携機構九州
株式会社エイシーエス

**PRIORITY
DOCUMENT**

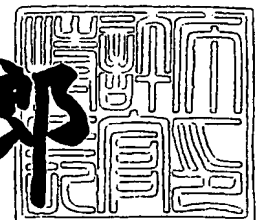
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

BEST AVAILABLE COPY

2003年 6月20日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田 信一郎



出証番号 出証特2003-3048407

【書類名】 特許願
 【整理番号】 H0301-01
 【あて先】 特許庁長官 殿
 【国際特許分類】 G06F 15/00
 H04L 9/32

【発明者】

【住所又は居所】 福岡県福岡市東区箱松4-22-14 松原寮A-30
 2

【氏名】 今本 健二

【発明者】

【住所又は居所】 東京都町田市三輪緑山1-3-2 緑山ヒルズ212

【氏名】 大河 克好

【特許出願人】

【識別番号】 800000035

【氏名又は名称】 株式会社 産学連携機構九州

【特許出願人】

【識別番号】 500196087

【氏名又は名称】 株式会社 エイシーエス

【代理人】

【識別番号】 100096862

【弁理士】

【氏名又は名称】 清水 千春

【電話番号】 03-3543-0036

【選任した代理人】

【識別番号】 100067046

【弁理士】

【氏名又は名称】 尾股 行雄

【電話番号】 03-3543-0036

【手数料の表示】

【予納台帳番号】 057761

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 ワンタイムIDの生成方法、認証方法、認証システム、サーバ、クライアントおよびプログラム

【特許請求の範囲】

【請求項1】

複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項2】

複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項3】

複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、

上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項 4】

一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項 5】

一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項 6】

一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、

一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするワンタイムIDの生成方法。

【請求項7】

請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、この生成したワンタイムIDと、当該第一装置に予め設定されたIDを少なくとも引数とする一方向関数Fcの関数値と、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、

上記第二装置が、上記ワンタイムIDおよび上記一方向関数Fcの関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムIDおよび一方向関数Fcの関数値との照合により、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたIDを少なくとも引数とする一方向関数Fsの関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、

上記第一装置が、上記一方向関数Fsの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数Fsの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

【請求項8】

上記一方向関数Fcとして、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるとともに、

上記一方向関数Fsとして、上記所定の共有鍵、上記Diffie-Hellman公開値の

一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるようにしたことを特徴とする請求項7に記載の認証方法。

【請求項9】

請求項2または請求項5に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムIDとして生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたID、上記第二装置に予め設定されたID、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第一のワンタイムIDに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムIDとして生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二のワンタイムIDを引数とする一方向関数 h の関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上

記第一装置に対して送信するステップと、

上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 h の関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数 h の関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とする認証方法。

【請求項10】

上記第二のワンタイムIDを生成する一方向関数として、上記第一のワンタイムIDを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とする請求項9に記載の認証方法。

【請求項11】

請求項3または請求項6に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数とを上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数とを上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記第二装置から受信した

上記第二のワンタイムIDとの比較により、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワンタイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

【請求項12】

請求項3または請求項6に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数を上記第二装置に対して送信するステップと、

上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして求め、この第二のワンタイムIDと上記第二の乱数を上記第一装置に対して送信するステップと、

上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワンタイムIDとの比較により、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタ

イムIDを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワнтаイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とする認証方法。

【請求項13】

上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とする請求項11または請求項12に記載の認証方法。

【請求項14】

上記第二装置が上記第二のワнтаイムIDと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワнтаイムIDとともに用いることを特徴とする請求項11～請求項13の何れかに記載の認証方法。

【請求項15】

上記第一装置が上記第三のワнтаイムIDを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワнтаイムIDとともに用いることを特徴とする請求項11～請求項14の何れかに記載の認証方法。

【請求項16】

請求項3または請求項6に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、

上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め

共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワнтаイムIDとして求め、当該第一装置に予め設定されたID、上記第二装置に予め設定されたIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワнтаイムIDとを上記第二装置に対して送信するステップと、

上記第二装置が、上記第一のワнтаイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワнтаイムIDとの照合により、上記第一装置を識別するステップと、

上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたIDおよび当該第二装置に予め設定されたIDに基づいて、上記第一装置の正当性を判定するステップと、

上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワнтаイムIDとして求め、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワнтаイムIDとを上記第一装置に対して送信するステップと、

上記第二装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、

上記第一装置が、上記第二のワнтаイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワнтаイムIDとの照合により、上記第二装置を識別するステップと、

上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたIDおよび当該第一装置に予め設定されたIDに基づいて、上記第二装置の正当性を判定するステップと、

上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とする認

証方法。

【請求項 17】

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とする請求項 16 に記載の認証方法。

【請求項 18】

請求項 1 または請求項 4 に記載のワンタイム ID の生成方法により生成されたワンタイム ID を用いてクライアントとの間で認証を行うサーバであって、

上記クライアントに予め設定されたクライアント ID を少なくとも引数とする一方向関数 F_c の関数値と、上記クライアントに予め記憶された Diffie-Hellman 公開値の一方と、上記ワンタイム ID とを上記クライアントから受信する受信手段と、

上記一方向関数の関数値 F_c および上記ワンタイム ID を演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイム ID および上記一方向関数 F_c の関数値との比較により、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバ ID を少なくとも引数とする一方向関数 F_s の関数値と、当該サーバに予め記憶された Diffie-Hellman 公開値の他方とを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

【請求項 19】

請求項 1 または請求項 4 に記載のワンタイム ID の生成方法により生成されたワンタイム ID を用いてサーバとの間で認証を行うクライアントであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイム ID を生成するとともに、当該クライアントに予め設定されたクライアント ID を少なくとも引数とする一方向関数 F_c の関数値を演算により求め、これらワンタイム ID および一方向関数 F_c の関数値と、当該クライアントに予め記憶された Diffie-Hellman 公開値の一方とを上記サーバに対して送信する送信手段と、

上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、

上記一方向関数 F_s の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 F_s の関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント。

【請求項20】

請求項18に記載のサーバと、請求項19に記載のクライアントとを備えてなることを特徴とする認証システム。

【請求項21】

請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムIDに基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、

上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 F_c の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する処理と、

上記一方向関数の関数値 F_c および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 F_c の関数値との比較により、上記クライアントの正当性を判定する処理と、

上記クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするプログラム。

【請求項22】

請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムIDに基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、

上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワнтаイムIDを生成するとともに、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数F_cの関数値を演算により求め、これらワнтаイムIDおよび一方向関数F_cの関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、

上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数F_sの関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、

上記一方向関数F_sの関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数F_sの関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするプログラム。

【請求項23】

請求項2または請求項5に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いてクライアントとの間で認証を行うサーバであって、

上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワнтаイムIDとして、この第一のワнтаイムID、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバID、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワнтаイムIDとを上記クライアントから受信する受信手段と、

上記第一のワнтаイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワнтаイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントID、上記サーバIDおよび上記第一のワнтаイムIDに基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の

可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワнтаイムIDとして生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントID、上記サーバIDおよび上記第二のワнтаイムIDを引数とする一方向関数 h の関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワнтаイムIDとを上記クライアントに対して送信する送信手段とを備えることを特徴とするサーバ。

【請求項24】

請求項2または請求項5に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いてサーバとの間で認証を行うクライアントであって、

上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワнтаイムIDとして生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバID、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワнтаイムIDを暗号化し、この暗号化データと上記第一のワнтаイムIDとを上記サーバに対して送信する送信手段と、

上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワнтаイムIDとし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワнтаイムID、上記第二の可変共有鍵、上記クライアントIDおよび上記サーバIDを引数とする一方向関数 h の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方と、上記第二のワнтаイムIDとを上記サーバから受信する受信手段と、

上記第二のワнтаイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワнтаイムIDとの照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記Diffie-Hellman公開値の他方と当該クライアントに予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、こ

の第二の可変共有鍵を用いて上記一方向関数 h の関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数 h の関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするクライアント

【請求項 25】

請求項 23 に記載のサーバと、請求項 24 に記載のクライアントとを備えてなることを特徴とする認証システム。

【請求項 26】

請求項 3 または請求項 6 に記載のワンタイム ID の生成方法により生成されたワンタイム ID を用いてクライアントとの間で相互に認証を行うサーバであって

上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイム ID として、この第一のワンタイム ID と、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイム ID として求め、この第二のワンタイム ID と上記第二の乱数とを上記クライアントに対して送信する送信手段と

上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワンタイム ID として、この第三のワンタイム ID を上記クライアントから受信する第二受信手段と、

上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワンタイム ID を演算により求め、この演算結果と上記クライアントから受信した上記第三のワンタイム ID との比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

【請求項 27】

請求項 3 または請求項 6 に記載のワンタイム ID の生成方法により生成された

ワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、この第一のワンタイムIDと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワンタイムIDとして、この第二のワンタイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワンタイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワンタイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワンタイムIDとして求め、この第三のワンタイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

【請求項28】

請求項26に記載のサーバと、請求項27に記載のクライアントとを備えてなることを特徴とする認証システム。

【請求項29】

請求項3または請求項6に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって

上記クライアントとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワンタイムIDとして、この第一のワンタイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、

第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする

一方向関数の関数値を第二のワнтаイムIDとして求め、この第二のワнтаイムIDと上記第二の乱数を上記クライアントに対して送信する送信手段と、

上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムIDとして、この第三のワнтаイムIDを上記クライアントから受信する第二受信手段と、

上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワнтаイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするサーバ。

【請求項 3 0】

請求項 3 または請求項 6 に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いてサーバとの間で相互に認証を行うクライアントであって

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムIDとして求め、この第一のワнтаイムIDと上記第一の乱数を上記サーバに対して送信する第一送信手段と、

上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムIDとして、この第二のワнтаイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、

上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムIDとの比較により、上記サーバの正当性を判定する判定手段と、

上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムIDとして求め、この第三のワнтаイムIDを上記サーバに対して送信する第二送信手段とを備えることを特徴とするクライアント。

【請求項 3 1】

請求項 2 9 に記載のサーバと、請求項 3 0 に記載のクライアントとを備えてな

ることを特徴とする認証システム。

【請求項 3 2】

請求項 3 または請求項 6 に記載のワンタイム ID の生成方法により生成されたワンタイム ID を用いてクライアントとの間で相互に認証を行うサーバであって

上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイム ID として、この第一のワンタイム ID を上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアント ID、当該サーバに予め設定されたサーバ ID を上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、

上記第一のワンタイム ID を演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイム ID との照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアント ID および上記サーバ ID に基づいて、上記クライアントの正当性を判定する判定手段と、

上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイム ID として求め、上記クライアント ID、上記サーバ ID および上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイム ID とを上記クライアントに対して送信する送信手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするサーバ。

【請求項 3 3】

請求項 3 または請求項 6 に記載のワンタイム ID の生成方法により生成されたワンタイム ID を用いてサーバとの間で相互に認証を行うクライアントであって

第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムIDとして求め、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、

上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムIDとして、この第二のワンタイムIDを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントIDおよび上記サーバIDを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、

上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバIDおよび上記クライアントIDに基づいて、上記サーバの正当性を判定する判定手段と、

上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするクライアント。

【請求項34】

請求項32に記載のサーバと、請求項33に記載のクライアントとを備えてなることを特徴とする認証システム。

【請求項35】

上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とする請求項34に記載の認証システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、複数の装置間またはアプリケーション間における認証に用いて好適なワンタイムIDの生成方法、上記ワンタイムIDを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムに関するものである。

【0002】

【従来の技術】

従来より、ネットワークを介してコンピュータ間（例えば、クライアント・サーバ間）で通信を行う際には、不正なアクセス等を排除するために、サービス等の提供に先立って認証が行われる。この認証においては、第三者が知り得ない所定の秘密情報（例えば、IDやパスワード、乱数、或いはそれら情報を引数とする関数値など）を予め双方が共有し、その秘密情報に基づいて各々の正当性を相互に検証するのが一般的である。

【0003】

他方、IETF (Internet Engineering Task Force) が公式に発行するRFC (Request For Comments) においては、インターネットでIPパケットの暗号化と認証を行なうセキュリティプロトコルとして、IPsec (Security Architecture for Internet Protocol) が規定されている。このIPsecでは、暗号・認証のパラメータを動的に生成して交換するIKE (Internet Key Exchange) という自動鍵交換のプロトコルが標準的に採用されている（例えば、特許文献1参照）。

そして、近年では、このIKEの方式にワンタイムIDを導入して、既知共有鍵を用いたIKEの方式で問題となっていた、ID情報保護、DoS (Denial of Service attack) 攻撃防止、リモートアクセスなどを実現したP-SIGMAと呼ばれる鍵交換・認証方式が提案されている。

【0004】

このP-SIGMAにおいては、例えば、図13に示すような手順で鍵交換および認証が行われている。

まず、クライアントが、SA (Security Association) の提案、乱数 R_c 、DH (Diffie-Hellman) 公開値 g^x 、OID (ワンタイムID) をサーバに対して

送信する。なお、SAの提案には、暗号アルゴリズムや認証方式、鍵交換に使用するパラメータ等に関する提案が含まれている。

【0005】

次いで、サーバが、受け取ったOIDからクライアントを識別し、識別できない場合には、通信を拒否する。識別できる場合には、受諾したSA、乱数 R_s 、DH公開値 g^y 、HASHs、セッション鍵 e で暗号化したIDs（サーバID）をクライアントに対して送信する。なお、セッション鍵 e は、既知共有鍵、乱数 R_s 、乱数 R_c およびDH共通鍵 g^{xy} を引数とする鍵付きハッシュ関数の関数値であり、HASHsは、既知共有鍵、乱数 R_s 、乱数 R_c 、DH公開値 g^x 、 g^y およびIDsを引数とする疑似乱数関数の関数値である。

【0006】

次いで、クライアントが、受け取ったHASHsを検証し、このHASHsに基づいてサーバの正当性を確認する。HASHsが正しければ、HASHc、セッション鍵 e で暗号化したIDc（クライアントID）をサーバに対して送信する。ここで、HASHcは、既知共有鍵、乱数 R_s 、乱数 R_c 、DH公開値 g^x 、 g^y およびIDcを引数とする疑似乱数関数の関数値である。

次いで、サーバが、受け取ったHASHcを検証し、このHASHcに基づいてクライアントの正当性を確認する。HASHcが正しければ、当該プロトコルを終了する。

【0007】

このP-SIGMAにおいて、OID（ワンタイムID）は、次のように定義されている。

【0008】

【数1】

$$OID_1 = \text{prf}(K, 1)$$

$$OID_2 = \text{prf}(K, 2)$$

...

$$OID_n = \text{prf}(K, n)$$

【0009】

この定義式において、 OID_n は n 番目の SA 確立時に用いられるワンタイム ID、 prf は疑似乱数関数、 K は既知共有鍵、若しくは既知共有鍵から生成された値である。

【0010】

このため、上記 P-SIGMA によれば、OID を導入したことにより、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であれば OID を識別情報として把握できるといった効果が得られるとともに、クライアント・サーバ間で通信が行われる度（すなわち、SA の生成または更新毎）に OID が変更されるため、第三者が次回の OID を予測することができないといった効果が得られる。

【0011】

【特許文献 1】

特開 2002-374238 号公報（段落番号 0002～0009）

【0012】

【発明が解決しようとする課題】

しかしながら、上記 P-SIGMA においては、既知共有鍵がひとたび知られてしまうと、すべての OID が予測されてしまい、その結果、OID の将来にわたる安全性（すなわち、PFS: Perfect Forward Security）を保証できなくなるといった問題点があった。

【0013】

以上、具体例として、P-SIGMA と呼ばれる鍵交換・認証方式について述べてきたが、一般に、ワンタイム ID を用いて複数の装置間またはアプリケーション間における認証を行う認証方式では、特定の秘密情報に基づいてすべてのワンタイム ID の生成を行っており、上記同様の問題点を有している。

【0014】

本発明は、かかる事情に鑑みてなされたもので、盗聴が困難で安全性に優れたワンタイム ID の生成方法、上記ワンタイム ID を用いた認証方法、認証システム、サーバ、クライアントおよびプログラムを提供することを目的とする。

【0015】

【課題を解決するための手段】

請求項 1 に記載の発明は、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイム ID として、当該ワンタイム ID を生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイム ID を生成するようにしたことを特徴とするものである。

【0016】

ここで、一方向関数とは、引数から結果（関数値）を求めるのは簡単であるが、結果から引数を求めるのは難しい関数のことを言い、この一方向関数には、例えば、ハッシュ関数、疑似乱数関数などが含まれる。

所定の通信単位としては、例えば、IPsec において SA が確立されてから当該 SA が無効になるまでの間にクライアント・サーバ間で行われる一連の通信を、所定の通信単位として設定することも可能であるし、装置間またはアプリケーション間で行われる 1 回のデータ送受信を所定の通信単位として設定することも可能である。

可変共有鍵は、上記所定の通信単位毎に変化し、且つ認証を行う装置間またはアプリケーション間で共有される、第三者が知り得ない秘密情報であれば、如何なる鍵であってもよい。

【0017】

認証とは、一方の装置（または一方のアプリケーション）が他方の装置（または他方のアプリケーション）にアクセスする際に、他方の装置が一方の装置の正当性を確認することを言い、識別情報とは、上記認証において少なくとも一方の装置から他方の装置に送信されて当該他方の装置が一方の装置を識別するのに用いる情報（ID）のことを云う。

また、上記認証には、一方の装置が他方の装置の認証を行う一方向認証と、双方の装置で相互に認証を行う相互認証とが含まれる。例えば、上記認証においてワンタイム ID を使用する方法としては、双方の装置でワンタイム ID を生成す

るとともに、一方の装置が他方の装置にワンタイムIDを送信し、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、一方の装置を識別或いは認証する方法が挙げられる。

【0018】

請求項2に記載の発明は、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

【0019】

請求項3に記載の発明は、複数の装置間またはアプリケーション間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法であって、上記認証を行う装置またはアプリケーションの各々において、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

【0020】

請求項4に記載の発明は、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

【0021】

請求項5に記載の発明は、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

【0022】

請求項6に記載の発明は、一方の装置と他方の装置間における認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを双方の装置で生成するとともに、一方の装置が他方の装置にワンタイムIDを送信して、他方の装置が、一方の装置から受信したワンタイムIDと自らが生成したワンタイムIDとの比較・照合により、他方の装置を識別或いは認証する場合において、一方の装置および他方の装置がワンタイムIDを生成する方法であって、一方の装置および他方の装置は、上記認証が必要な所定の通信単位内で乱数を生成するとともに、この乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにしたことを特徴とするものである。

【0023】

請求項7に記載の発明は、請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムID (SIGNAL_n) を用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、この生成したワンタイムIDと、当該第一装置に予め設定されたIDを少なくとも引数とする一方向関数Fcの関数値と、当該第一装置に予め記

憶されたDiffie-Hellman公開値の一方とを上記第二装置に対して送信するステップと、上記第二装置が、上記ワンタイムIDおよび上記一方向関数Fcの関数値を演算により求め、この演算結果と、上記第一装置から受信したワンタイムIDおよび一方向関数Fcの関数値との照合により、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、当該第二装置に予め設定されたIDを少なくとも引数とする一方向関数Fsの関数値と、当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とを上記第一装置に対して送信するステップと、上記第一装置が、上記一方向関数Fsの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数Fsの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

【0024】

請求項8に記載の発明は、請求項7に記載の認証方法において、上記一方向関数Fcとして、所定の共有鍵、上記Diffie-Hellman公開値の一方、上記第一装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるとともに、上記一方向関数Fsとして、上記所定の共有鍵、上記Diffie-Hellman公開値の一方、上記Diffie-Hellman公開値の他方、上記第二装置に予め設定されたID、上記ワンタイムIDを引数とする疑似乱数関数を用いるようにしたことを特徴とするものである。

【0025】

請求項9に記載の発明は、請求項2または請求項5に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、上記第二装置との間で予め共有化された第一の可変共有鍵と当該第一装置の通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムID (SIGNAL_{n,j}) として生成するとともに、上記第一の可変共有鍵を用いて、当該第一装置に予め設定されたID、上記第二装置に予め設定されたID、当該第一装置に予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記第二装置に対して送信するス

テップと、上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第一のワンタイムIDに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、上記第一の可変共有鍵と当該第二装置の通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムID (SIGNAL'_{n,1}) として生成するとともに、上記第一装置から受信したDiffie-Hellman公開値の一方と当該第二装置に予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二のワンタイムIDを引数とする一方向関数hの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記第一装置に対して送信するステップと、上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記第二装置から受信した上記Diffie-Hellman公開値の他方と当該第一装置に予め記憶された上記Diffie-Hellman公開値の一方とからDiffie-Hellman共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数hの関数値を演算により求め、この演算結果と、上記第二装置から受信した一方向関数hの関数値との照合により、上記第二装置の正当性を判定するステップとを有することを特徴とするものである。

【0026】

請求項10に記載の発明は、請求項9に記載の認証方法において、上記第二のワンタイムIDを生成する一方向関数として、上記第一のワンタイムIDを生成する一方向関数とは異なる一方向関数を用いるようにしたことを特徴とするものである。

【0027】

請求項11に記載の発明は、請求項3または請求項6に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID（ $SIGNAL_{c1}$ ）として求め、この第一のワнтаイムIDと上記第一の乱数とを上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID（ $SIGNAL_{s1}$ ）として求め、この第二のワнтаイムIDと上記第二の乱数とを上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワнтаイムIDとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムID（ $SIGNAL_{c2}$ ）として求め、この第三のワнтаイムIDを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワнтаイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

【0028】

請求項12に記載の発明は、請求項3または請求項6に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いて、第一装置と第二装置間における認証（相互認証）を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID（ $SIGNAL_{c1}$ ）として求め、

この第一のワнтаイムIDと上記第一の乱数を上記第二装置に対して送信するステップと、上記第二装置が、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID (SIGNAL_{s1}) として求め、この第二のワнтаイムIDと上記第二の乱数を上記第一装置に対して送信するステップと、上記第一装置が、上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記第二装置から受信した上記第二のワнтаイムIDとの比較により、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムID (SIGNAL_{c2}) として求め、この第三のワнтаイムIDを上記第二装置に対して送信するステップと、上記第二装置が、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記第一装置から受信した上記第三のワнтаイムIDとの比較により、上記第一装置の正当性を判定するステップとを有することを特徴とするものである。

【0029】

請求項13に記載の発明は、請求項11または請求項12に記載の認証方法において、上記第一の乱数と上記第二の乱数を、上記第一装置と上記第二装置との間で予め共有化された共有鍵で暗号化した状態で、送信するようにしたことを特徴とするものである。

【0030】

請求項14に記載の発明は、請求項11～請求項13の何れかに記載の認証方法において、上記第二装置が上記第二のワнтаイムIDと上記第二の乱数とを上記第一装置に対して送信するステップにおいて、上記第二装置は、上記第一装置との間で予め共有化された乱数を初期乱数として、この初期乱数と上記第一の乱数を引数とする所定の演算を行い、この演算結果を上記第一装置に対して送信する一方、上記第一装置は、上記第二装置の正当性の判定材料として、上記第二装置から受信した上記演算結果を、上記第二のワнтаイムIDとともに用いることを特徴とするものである。

【0031】

請求項15に記載の発明は、請求項11～請求項14の何れかに記載の認証方法において、上記第一装置が上記第三のワンタイムIDを上記第二装置に対して送信するステップにおいて、上記第一装置は、上記第一の乱数と上記第二の乱数を引数とする所定の演算を行い、この演算結果を上記第二装置に対して送信する一方、上記第二装置は、上記第一装置の正当性の判定材料として、上記第一装置から受信した上記演算結果を、上記第三のワンタイムIDとともに用いることを特徴とするものである。

【0032】

請求項16に記載の発明は、請求項3または請求項6に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、第一装置と第二装置間における認証を行う認証方法であって、上記第一装置が、第一の乱数を生成するとともに、上記第二装置との間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID ($SIGNAL_{ci}$) として求め、当該第一装置に予め設定されたID、上記第二装置に予め設定されたIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記第二装置に対して送信するステップと、上記第二装置が、上記第一のワンタイムIDを演算により求め、この演算結果と、上記第一装置から受信した上記第一のワンタイムIDとの照合により、上記第一装置を識別するステップと、上記第二装置が、上記第一装置を識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる、上記第一装置に予め設定されたIDおよび当該第二装置に予め設定されたIDに基づいて、上記第一装置の正当性を判定するステップと、上記第二装置が、上記第一装置を正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID ($SIGNAL_{si}$) として求め、上記第一装置に予め設定されたID、当該第二装置に予め設定されたIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムIDとを上記第一装置に対して送信するステップと、上記第二装置

が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップと、上記第一装置が、上記第二のワンタイムIDを演算により求め、この演算結果と、上記第二装置から受信した上記第二のワンタイムIDとの照合により、上記第二装置を識別するステップと、上記第一装置が、上記第二装置を識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる、上記第二装置に予め設定されたIDおよび当該第一装置に予め設定されたIDに基づいて、上記第二装置の正当性を判定するステップと、上記第一装置が、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換するステップとを有することを特徴とするものである。

【0033】

請求項17に記載の発明は、請求項16に記載の認証方法において、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後に、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようにしたことを特徴とするものである。

【0034】

請求項18に記載の発明は、請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムID ($SIGNAL_n$) を用いてクライアントとの間で認証を行うサーバであって、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 F_c の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する受信手段と、上記一方向関数の関数値 F_c および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 F_c の関数値との比較により、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、当該サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、当該サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する送信

手段とを備えることを特徴とするものである。

【0035】

請求項19に記載の発明は、請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムID ($SIGNAL_n$) を用いてサーバとの間で認証を行うクライアントであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワンタイムIDを生成するとともに、当該クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 F_c の関数値を演算により求め、これらワンタイムIDおよび一方向関数 F_c の関数値と、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する送信手段と、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する受信手段と、上記一方向関数 F_s の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 F_s の関数値との比較により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

【0036】

請求項20に記載の本発明に係る認証システムは、請求項18に記載のサーバと、請求項19に記載のクライアントとを備えてなることを特徴とするものである。

【0037】

請求項21に記載の発明は、請求項1または請求項4に記載のワンタイムIDの生成方法により生成されたワンタイムID ($SIGNAL_n$) に基づいてクライアントとの間で認証を行うサーバに実行させるプログラムであって、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 F_c の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方と、上記ワンタイムIDとを上記クライアントから受信する処理と、上記一方向関数の関数値 F_c および上記ワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記ワンタイムIDおよび上記一方向関数 F_c の関数値との比較により、上記クライアントの正当性を判定する処理と、上記

クライアントが正当であると判定された場合に、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記クライアントに対して送信する処理とを上記サーバに実行させることを特徴とするものである。

【0038】

請求項22に記載の発明は、請求項1または請求項4に記載のワнтаイムIDの生成方法により生成されたワнтаイムID ($SIGNAL_n$) に基づいてサーバとの間で認証を行うクライアントに実行させるプログラムであって、上記サーバとの間で予め共有化された可変共有鍵を用いて上記ワнтаイムIDを生成するとともに、上記クライアントに予め設定されたクライアントIDを少なくとも引数とする一方向関数 F_c の関数値を演算により求め、これらワнтаイムIDおよび一方向関数 F_c の関数値と、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方とを上記サーバに対して送信する処理と、上記サーバに予め設定されたサーバIDを少なくとも引数とする一方向関数 F_s の関数値と、上記サーバに予め記憶されたDiffie-Hellman公開値の他方とを上記サーバから受信する処理と、上記一方向関数 F_s の関数値を演算により求め、この演算結果と、上記サーバから受信した上記一方向関数 F_s の関数値との比較により、上記サーバの正当性を判定する処理とを上記クライアントに実行させることを特徴とするものである。

【0039】

請求項23に記載の発明は、請求項2または請求項5に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いてクライアントとの間で認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の可変共有鍵と上記クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワнтаイムID ($SIGNAL_{n,j}$) として、この第一のワнтаイムID、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバID、上記クライアントに予め記憶されたDiffie-Hellman公開値の一方を上記第一の可変共有鍵で暗号化した暗号化データと、上記第一のワнтаイムIDとを上記クライアントから受信する受信手段と、上記第一のワ

タイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記第一の可変共有鍵を用いて上記暗号化データを復号し、この復号したデータに含まれる、上記クライアントID、上記サーバIDおよび上記第一のワンタイムIDに基づいて、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、上記第一の可変共有鍵と当該サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムID (SIGNAL' $n, 1$) として生成するとともに、上記クライアントから受信したDiffie-Hellman公開値の一方と当該サーバに予め記憶されたDiffie-Hellman公開値の他方とからDiffie-Hellman共通鍵を第二の可変共有鍵として生成し、この第二の可変共有鍵、上記クライアントID、上記サーバIDおよび上記第二のワンタイムIDを引数とする一方向関数hの関数値と、上記Diffie-Hellman公開値の他方と、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段とを備えることを特徴とするものである。

【0040】

請求項24に記載の発明は、請求項2または請求項5に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いてサーバとの間で認証を行うクライアントであって、上記サーバとの間で予め共有化された第一の可変共有鍵と当該クライアントの通信順序に関する情報とを引数とする一方向関数の関数値を第一のワンタイムID (SIGNAL n, j) として生成するとともに、上記第一の可変共有鍵を用いて、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバID、当該クライアントに予め記憶されたDiffie-Hellman公開値の一方および上記第一のワンタイムIDを暗号化し、この暗号化データと上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、上記第一の可変共有鍵と上記サーバの通信順序に関する情報とを引数とする一方向関数の関数値を第二のワンタイムID (SIGNAL' $n, 1$) とし、Diffie-Hellman共通鍵を第二の可変共有鍵として、上記第二のワンタイムID、上記第二の可変共有鍵、上記クライアントIDおよび上記サーバIDを引数とす

る一方向関数 h の関数値と、上記サーバに予め記憶された Diffie-Hellman 公開値の他方と、上記第二のワнтаイム ID とを上記サーバから受信する受信手段と、上記第二のワнтаイム ID を演算により求め、この演算結果と、上記サーバから受信した上記第二のワнтаイム ID との照合により、上記サーバを識別し、上記サーバを識別した場合に、上記サーバから受信した上記 Diffie-Hellman 公開値の他方と当該クライアントに予め記憶された上記 Diffie-Hellman 公開値の一方とから Diffie-Hellman 共通鍵を上記第二の可変共有鍵として生成するとともに、この第二の可変共有鍵を用いて上記一方向関数 h の関数値を演算により求め、この演算結果と、上記サーバから受信した一方向関数 h の関数値との照合により、上記サーバの正当性を判定する判定手段とを備えることを特徴とするものである。

【0041】

請求項 25 に記載の本発明に係る認証システムは、請求項 23 に記載のサーバと、請求項 24 に記載のクライアントとを備えてなることを特徴とするものである。

【0042】

請求項 26 に記載の発明は、請求項 3 または請求項 6 に記載のワнтаイム ID の生成方法により生成されたワнтаイム ID を用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワнтаイム ID ($SIGNAL_{c1}$) として、この第一のワнтаイム ID と、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するとともに、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワнтаイム ID ($SIGNAL_{s1}$) として求め、この第二のワнтаイム ID と上記第二の乱数とを上記クライアントに対して送信する送信手段と、上記第一の乱数、上記第二の乱数および第二の共有鍵を引数とする一方向関数の関数値を第三のワнтаイム ID ($SIGNAL_{c2}$) として、この第三のワнтаイム ID を上記クライアントから受信する第二受信手段と、上記第一の乱数および上記第二の乱数に基づいて上記第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数に基づいて上記第三のワнтаイム

ＩＤを演算により求め、この演算結果と上記クライアントから受信した上記第三のワнтаイムＩＤとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

【0043】

請求項２７に記載の発明は、請求項３または請求項６に記載のワнтаイムＩＤの生成方法により生成されたワнтаイムＩＤを用いてサーバとの間で相互に認証を行うクライアントであって、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された第一の共有鍵を引数とする一方向関数の関数値を第一のワнтаイムＩＤ（ $SIGNAL_{c1}$ ）として求め、この第一のワнтаイムＩＤと上記第一の乱数とを上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記第一の共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムＩＤ（ $SIGNAL_{s1}$ ）として、この第二のワнтаイムＩＤと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記第一の共有鍵に基づいて上記第二のワнтаイムＩＤを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムＩＤとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数および上記第二の乱数に基づいて第二の共有鍵を生成するとともに、この第二の共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムＩＤ（ $SIGNAL_{c2}$ ）として求め、この第三のワнтаイムＩＤを上記サーバに対して送信する第二送信手段とを備えることを特徴とするものである。

【0044】

請求項２８に記載の本発明に係る認証システムは、請求項２・６に記載のサーバと、請求項２７に記載のクライアントとを備えてなることを特徴とするものである。

【0045】

請求項２９に記載の発明は、請求項３または請求項６に記載のワнтаイムＩＤの生成方法により生成されたワнтаイムＩＤを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有

鍵を引数とする一方向関数の関数値を第一のワнтаイムID ($SIGNAL_{c1}$) として、この第一のワнтаイムIDと、上記クライアントで生成された第一の乱数とを上記クライアントから受信する第一受信手段と、第二の乱数を生成するとともに、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID ($SIGNAL_{s1}$) として求め、この第二のワнтаイムIDと上記第二の乱数を上記クライアントに対して送信する送信手段と、上記共有鍵、上記第一の乱数および上記第二の乱数を引数とする一方向関数の関数値を第三のワнтаイムID ($SIGNAL_{c2}$) として、この第三のワнтаイムIDを上記クライアントから受信する第二受信手段と、上記第一の乱数、上記第二の乱数および上記共有鍵に基づいて上記第三のワнтаイムIDを演算により求め、この演算結果と上記クライアントから受信した上記第三のワнтаイムIDとの比較により、上記クライアントの正当性を判定する判定手段とを備えることを特徴とするものである。

【0046】

請求項30に記載の発明は、請求項3または請求項6に記載のワнтаイムIDの生成方法により生成されたワнтаイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵を引数とする一方向関数の関数値を第一のワнтаイムID ($SIGNAL_{c1}$) として求め、この第一のワнтаイムIDと上記第一の乱数を上記サーバに対して送信する第一送信手段と、上記第一の乱数と上記共有鍵とを引数とする一方向関数の関数値を第二のワнтаイムID ($SIGNAL_{s1}$) として、この第二のワнтаイムIDと、上記サーバで生成された第二の乱数とを上記サーバから受信する受信手段と、上記第一の乱数および上記共有鍵に基づいて上記第二のワнтаイムIDを演算により求め、この演算結果と上記サーバから受信した上記第二のワнтаイムIDとの比較により、上記サーバの正当性を判定する判定手段と、上記判定手段により上記サーバが正当であると判定された場合に、上記第一の乱数、上記第二の乱数および上記共有鍵を引数とする一方向関数の関数値を第三のワнтаイムID ($SIGNAL_{c2}$) として求め、この第三のワнтаイムIDを上記サーバに対して送信する第二送信手段とを備えることを特

徴とするものである。

【0047】

請求項31に記載の本発明に係る認証システムは、請求項29に記載のサーバと、請求項30に記載のクライアントとを備えてなることを特徴とするものである。

【0048】

請求項32に記載の発明は、請求項3または請求項6に記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いてクライアントとの間で相互に認証を行うサーバであって、上記クライアントとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID ($SIGNAL_{ci}$) として、この第一のワンタイムIDを上記クライアントから受信するとともに、上記クライアントで生成された第一の乱数、上記クライアントに予め設定されたクライアントID、当該サーバに予め設定されたサーバIDを上記共有鍵で暗号化した第一の暗号化データを上記クライアントから受信する受信手段と、上記第一のワンタイムIDを演算により求め、この演算結果と、上記クライアントから受信した上記第一のワンタイムIDとの照合により、上記クライアントを識別し、上記クライアントを識別できた場合に、上記共有鍵を用いて上記第一の暗号化データを復号し、この復号したデータに含まれる上記クライアントIDおよび上記サーバIDに基づいて、上記クライアントの正当性を判定する判定手段と、上記判定手段が上記クライアントを正当であると判定した場合に、第二の乱数を生成するとともに、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID ($SIGNAL_{si}$) として求め、上記クライアントID、上記サーバIDおよび上記第二の乱数を上記共有鍵で暗号化した第二の暗号化データと、上記第二のワンタイムIDとを上記クライアントに対して送信する送信手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

【0049】

請求項33に記載の発明は、請求項3または請求項6に記載のワンタイムID

の生成方法により生成されたワンタイムIDを用いてサーバとの間で相互に認証を行うクライアントであって、第一の乱数を生成するとともに、上記サーバとの間で予め共有化された共有鍵、第一の記憶乱数および第二の記憶乱数を引数とする一方向関数の関数値を第一のワンタイムID ($SIGNAL_{ci}$) として求め、当該クライアントに予め設定されたクライアントID、上記サーバに予め設定されたサーバIDおよび上記第一の乱数を上記共有鍵で暗号化した第一の暗号化データと、上記第一のワンタイムIDとを上記サーバに対して送信する送信手段と、上記第一の乱数、上記第二の記憶乱数および上記共有鍵を引数とする一方向関数の関数値を第二のワンタイムID ($SIGNAL_{si}$) として、この第二のワンタイムIDを上記サーバから受信するとともに、上記サーバで生成された第二の乱数、上記クライアントIDおよび上記サーバIDを上記共有鍵で暗号化した第二の暗号化データを上記サーバから受信する受信手段と、上記第二のワンタイムIDを演算により求め、この演算結果と、上記サーバから受信した上記第二のワンタイムIDとの照合により、上記サーバを識別し、上記サーバを識別できた場合に、上記共有鍵を用いて上記第二の暗号化データを復号し、この復号したデータに含まれる上記サーバIDおよび上記クライアントIDに基づいて、上記サーバの正当性を判定する判定手段と、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換する置換手段とを備えることを特徴とするものである。

【0050】

請求項34に記載の本発明に係る認証システムは、請求項32に記載のサーバと、請求項33に記載のクライアントとを備えてなることを特徴とするものである。

【0051】

請求項35に記載の発明は、請求項34に記載の認証システムにおいて、上記サーバおよび上記クライアントは、上記第一の記憶乱数を上記第一の乱数に、上記第二の記憶乱数を上記第二の乱数にそれぞれ置換した後で、これら第一の記憶乱数および第二の記憶乱数に基づいて上記共有鍵を生成することにより、当該共有鍵を変化させるようになっていることを特徴とするものである。

【0052】

請求項1または請求項4に記載の発明によれば、可変共有鍵を引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成するようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化することとなるので、漏れた可変共有鍵を用いて生成されたワнтаイムID以外のワнтаイムIDを予測することはできない。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

【0053】

請求項2または請求項5に記載の発明によれば、可変共有鍵と通信順序または回数に関する情報とを引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成するようにしたため、例えば、可変共有鍵が第三者に漏れたとしても、所定の通信単位毎に可変共有鍵が変化するとともに、各通信毎に通信順序または回数に関する情報も変化することとなるので、漏れた可変共有鍵を用いて生成されたワнтаイムID以外のワнтаイムIDを予測することは事実上不可能となり、また漏れた可変共有鍵を用いて生成されたワнтаイムIDの予測自体も非常に困難なものとなる。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

【0054】

請求項3または請求項6に記載の発明によれば、所定の通信単位内で生成された乱数と所定の共有鍵とを引数とする一方向関数の関数値を求め、この関数値からワнтаイムIDを生成するようにしたため、例えば、共有鍵が第三者に漏れたとしても、乱数によって一方向関数の関数値が所定の通信単位毎に変化することとなるので、所定の通信単位内で生成される乱数がわからない限り、ワнтаイムIDを予測することはできない。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

【0055】

請求項7～請求項35の何れかに記載の発明によれば、請求項1～請求項6の何れかに記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、装置間（クライアント・サーバ間）における認証を行うようにしたので、第三者（攻撃者）が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握することができる。

したがって、DOS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることができる。

【0056】

請求項8に記載の発明によれば、第一装置の正当性を判定するのに用いる一方向関数 F_c として、所定の共有鍵、Diffie-Hellman公開値の一方、第一装置に予め設定されたID、ワンタイムIDを引数とする疑似乱数関数を用いるとともに、第二装置の正当性を判定するのに用いる一方向関数 F_s として、所定の共有鍵、Diffie-Hellman公開値の一方、Diffie-Hellman公開値の他方、第二装置に予め設定されたID、ワンタイムIDを引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では3回必要であった通信回数を2回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

【0057】

【発明の実施の形態】

〔第1の実施形態〕

図1は、本発明に係る認証システムの一実施形態を示す概略構成図である。この認証システムは、公衆回線網やインターネット等のネットワーク40を介して相互に接続されたサーバ（第二装置）10とクライアント（第一装置）20とにより概略構成されている。この実施形態では、種々のサービスを提供する複数のサーバA、B、C、…がサーバ10に接続され、当該サーバ10が、サーバA、B、C、…へのアクセスの可否を決定する認証サーバとして機能するようになっている。

【0058】

サーバ10は、図2に示すように、CPU11、RAM12、記憶装置13、入力装置14、表示装置15および通信装置16等により構成され、各部はバス17により接続されている。

【0059】

CPU (Central Processing Unit) 11は、記憶装置13の記憶領域に格納されている各種処理プログラム、入力装置14や通信装置16から入力される各種指示、あるいは指示に対応する各種データ等をRAM12に格納し、それら入力指示および各種データに応じてRAM12に格納した各種処理プログラムに従って各種処理を実行し、その処理結果をRAM12に一時的に記憶するとともに、表示装置15等に出力する。

【0060】

このCPU11は、当該サーバ10における受信手段および判定手段を構成しており、クライアントIDを引数とする一方向関数（一方向関数 F_c ）の関数値であるHASH c 、ワンタイムID (SIGNAL)、DH公開値 g^x (Diffie-Hellman公開値の一方) をクライアント20から受信した場合（すなわち、クライアント20からアクセスの要求を受けた場合）に、クライアント20から受信した受信データと記憶装置13に記憶されている記憶データを用いてワンタイムIDおよびHASH c を演算により求め、この演算結果と、クライアント20から受信したワンタイムIDおよびHASH c との比較により、クライアント20の正当性を判定する処理を実行する。

【0061】

また、CPU11は、当該サーバ10における送信手段を構成しており、クライアント20が正当であると判定される場合に、上記受信データおよび上記記憶データを用いて、サーバIDを引数とする一方向関数（一方向関数 F_s ）の関数値であるHASH s を演算により求め、このHASH s と、記憶装置13に記憶されているDH公開値 g^y (Diffie-Hellman公開値の他方) とをクライアント20に対して送信する処理を実行する。

【0062】

なお、上記ワンタイムID (SIGNAL) は、サーバ・クライアント間における認証において一回限り使用可能な識別情報であり、このワンタイムIDを生成する場合には、所定の通信単位毎に変化する暗号化鍵K (可変共有鍵) を記憶装置13から読み込んで、この暗号化鍵Kを引数とするハッシュ関数 (一方向関数) の関数値を求め、この関数値から上記ワンタイムIDを生成する。

【0063】

RAM (Random Access Memory) 12は、クライアント20等との間で送受信されるデータなど、認証に関する各種データを一時記憶する記憶領域や、CPU11の作業領域などを備えている。

【0064】

記憶装置13は、プログラムやデータ等が記憶される記憶媒体13aを有し、この記憶媒体13aは磁氣的、光学的記録媒体、若しくは半導体メモリで構成されている。この記憶媒体13aは記憶装置13に固定的に設けたもの、若しくは着脱自在に装着するものであり、CPU11により実行される各種処理プログラムや制御データ等を記憶する記憶領域、認証に関する各種データ (例えば、クライアント20やID発行管理サーバ30 (後述) から取得したデータ、認証の処理過程で生成されたデータなど) を格納する記憶領域などを備えている。なお、この記憶媒体13aに記憶するプログラムやデータなどは、その一部若しくは全部を他のサーバ等からネットワーク40を介して受信して記憶する構成とすることも可能である。この記憶媒体13aには、サーバID、DH公開値 g^y 、クライアント20との間で共有化された乱数Rなどが、認証処理を開始する前段階で予め格納された状態となっている。

【0065】

入力装置14は、キーボードやポインティングデバイス等により構成され、入力指示信号をCPU11に対して出力する。

表示装置15は、CRT (Cathode Ray Tube) やLCD (Liquid Crystal Display) 等により構成され、CPU11から入力される表示データを表示する。

通信装置16は、モデムやルータ、ブリッジ等により構成され、ネットワーク40を介してクライアント20等より受信したデータをCPU11に出力すると

ともに、CPU 11より受信したデータをネットワーク40を介してクライアント20等に対して出力する。

【0066】

一方、クライアント20は、図3に示すように、CPU 21、RAM 22、記憶装置23、入力装置24、表示装置25および通信装置26等により構成され、各部はバス27により接続されている。具体的に、クライアント20としては、例えば、パーソナルコンピュータや、PDA (Personal Digital Assistance) 等の携帯情報端末、インターネット接続サービスを利用可能な携帯電話などが挙げられる。なお、このクライアント20の各構成要素は、前述したサーバ10の各構成要素とほぼ同様であるので、相違点のみを以下に説明する。

【0067】

すなわち、クライアント20のCPU 21は、当該クライアント20における送信手段を構成しており、入力装置24からの指示入力等に基づいて、ワンタイムID (SIGNAL) を生成するとともに、クライアントIDを引数とする一方向関数 (一方向関数 F_c) の関数値であるHASH c を求め、これらワンタイムIDおよびHASH c と、記憶装置23に予め記憶されたDH公開値 g^x (Diffie-Hellman公開値の一方) とをサーバ10に対して送信する処理を実行する。

【0068】

また、CPU 21は、当該クライアント20における受信手段および判定手段を構成しており、サーバIDを引数とする一方向関数 (一方向関数 F_s) の関数値であるHASH s と、DH公開値 g^y (Diffie-Hellman公開値の他方) とをサーバ10から受信した場合 (すなわち、サーバ10によってクライアント20が正当であると判定された場合) に、サーバ10から受信した受信データと記憶装置23に記憶されている記憶データを用いてHASH s を演算により求め、この演算結果と、サーバ10から受信したHASH s との比較により、サーバ10の正当性を判定する処理を実行する。

【0069】

記憶装置23は、プログラムやデータ等が記憶される記憶媒体23aを有し、この記憶媒体23aは、上記CPU 21により実行される各種処理プログラムや

制御データ等を記憶する記憶領域、認証に関する各種データ（例えば、サーバ10やID発行管理サーバ30（後述）から取得したデータ、認証の処理過程で生成されたデータなど）を格納する記憶領域などを備えている。この記憶媒体23aには、クライアントID、DH公開値 g^x 、サーバ10との間で共有化された乱数Rなどが、認証処理を開始する前段階で予め格納された状態となっている。

【0070】

ID発行管理サーバ30は、クライアント・サーバ間で共有化される秘密情報（例えば、ワンタイムIDの初期値を生成するのに用いられる乱数Rなど）や、クライアントID、サーバIDなどを発行・管理するためのサーバである。このID発行管理サーバ30は、クライアント20を利用するユーザのID（例えば、クレジットNo、住基ネットID、社員No、学生No、特定会員Noなど）に上記秘密情報やパスワードなどを対応付けた状態で格納するデータベースを有している。また、ID発行管理サーバ30は、一定の周期で上記データベース内の秘密情報を更新し、この更新した秘密情報をオンライン（例えば、電子メールなど）またはオフライン（例えば、郵送など）で、クライアント20とサーバ10の双方に配布するようになっている。なお、上記秘密情報の発行は、クライアント20またはサーバ10からの発行依頼に基づくものであってもよい。

【0071】

次に、上記構成からなる認証システムによって行われる認証方法の第1の実施形態について、図4に基づいて説明する。この方法は、RFC2409において規定されたIKEの方式に、本発明に係るワンタイムID（SIGNAL）を適用したものである。

【0072】

まず、ステップS1では、IKEによるSA生成に際してイニシエータとなるクライアント20が、ワンタイムID（SIGNAL）を生成するとともに、HASHcを演算により求め、これらワンタイムIDおよびHASHcと、記憶装置23に記憶されたDH公開値 g^x とをSAの提案とともに、レスポндаとなるサーバ10に対して送信する処理を実行する。

【0073】

ここで、ワンタイムIDであるSIGNALは、例えば、ハッシュ関数を用いて、次のように生成される。

【0074】

【数2】

$$\text{SIGNAL}_1 = R$$

$$\text{SIGNAL}_2 = \text{hash}(K_1)$$

$$\text{SIGNAL}_3 = \text{hash}(K_2)$$

...

$$\text{SIGNAL}_n = \text{hash}(K_{n-1})$$

【0075】

上記SIGNALの定義式において、hashはハッシュ関数、RはID発行管理サーバ30からサーバ10とクライアント20の双方に発行されて両者間で共有化された乱数、 K_i はi番目のセッションで生成されたサーバ・クライアント共有の暗号化鍵（可変共有鍵）である。なお、上記セッションは、SAを確立してから当該SAが無効になるまでの通信単位を示している。

【0076】

すなわち、上記SIGNALの定義式によれば、前回のセッションで生成された上記暗号化鍵Kを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのSIGNALとして用いるようにしている。また、最初のセッションでは、サーバ・クライアント間で予め共有化された乱数RをSIGNALの初期値として用いるようにしている。また、上記暗号化鍵 K_i は、例えば、次式(1)から求められる。

【0077】

$$K_i = \text{prf}(\text{共有鍵}, g^{xy}, \text{SIGNAL}_i) \quad \dots (1)$$

【0078】

この式(1)において、 g^{xy} はDH共通鍵であり、共有鍵は、サーバ・クライアント間の任意の共有鍵である。

【0079】

一方、HASHcは、次式(2)に示すように、共有鍵、DH公開値 g^x 、I

ID_c (クライアントID) および $SIGNAL$ を引数とする疑似乱数関数 (鍵付きハッシュ関数) の関数値として求められる。

【0080】

$$HASH_c = \text{prf}(\text{共有鍵}, g^x, ID_c, SIGNAL) \quad \cdots (2)$$

【0081】

次いで、ステップS2では、サーバ10が、 $SIGNAL$ と $HASH_c$ を演算により求め、これら演算結果と、クライアント20から受信した $SIGNAL$ および $HASH_c$ との比較により、クライアント20の正当性を判定する処理を実行する。

【0082】

上記判定の結果、受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、 $HASH_s$ を演算により求め、この $HASH_s$ と、記憶装置13に記憶されているDH公開値 g^y とを、受諾したSAとともにクライアント20に対して送信する処理を実行する (ステップS3)。一方、受信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0083】

ここで、 $HASH_s$ は、次式(3)に示すように、共有鍵、DH公開値 g^x 、 g^y 、 ID_s (サーバID) および $SIGNAL$ を引数とする疑似乱数関数 (鍵付きハッシュ関数) の関数値として求められる。

【0084】

$$HASH_s = \text{prf}(\text{共有鍵}, g^x, g^y, ID_s, SIGNAL) \quad \cdots (3)$$

【0085】

また、このステップS3においては、記憶装置13に記憶されているDH公開値 g^y と、クライアント20から受信したDH公開値 g^x とからDH共通鍵 g^{xy} を生成して、DH共通鍵 g^{xy} を記憶装置13に格納する処理も併せて行う。

【0086】

次いで、ステップS4では、クライアント20が、 $HASH_s$ を演算により求

め、この演算結果と、サーバ10から受信したHASHsとの比較により、サーバ10の正当性を判定する処理を実行する。

【0087】

上記判定の結果、受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、記憶装置23に記憶されているDH公開値 g^x と、サーバ10から受信したDH公開値 g^y とからDH共通鍵 g^{xy} を生成して記憶装置23に格納した後、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

【0088】

以上のように、この第1の実施形態によれば、セッション毎に変化する暗号化鍵K（可変共有鍵）を引数とするハッシュ関数の関数値をワнтаイムID（SIGNAL）として用いるようにしたので、例えば、暗号化鍵Kが第三者に漏れたとしても、セッション毎に暗号化鍵Kが変化することとなるので、漏れた暗号化鍵Kを用いて生成されたワнтаイムID以外のワнтаイムIDを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（PFS）を実現することが可能になる。

【0089】

また、上記ワнтаイムID（SIGNAL）を用いて、クライアント・サーバ間における認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワнтаイムIDを識別情報として把握することができる。

したがって、DoS攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

【0090】

また、この実施形態では、クライアント20の正当性を判定するのに用いる一

方向関数 F_c として、共有鍵、DH公開値 g^x 、 ID_c (クライアントID) および $SIGNAL$ を引数とする疑似乱数関数を用いるとともに、サーバ10の正当性を判定するのに用いる一方向関数 F_s として、共有鍵、DH公開値 g^x 、 g^y 、 ID_s (サーバID) および $SIGNAL$ を引数とする疑似乱数関数を用いるようにしたので、従来の鍵交換・認証方式では3回必要であった通信回数を2回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

【0091】

〔第2の実施形態〕

前述した第1の実施形態では、前回のセッションで生成された暗号化鍵(可変共有鍵)を引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのワンタイムID($SIGNAL$)として用いるようにしたが、この第2の実施形態では、前回のセッションで生成された共有鍵と、当該セッションにおける通信順序とを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションの各通信時におけるワンタイムIDとして用いるようにしている。この第2の実施形態特有の部分以外は、第1の実施形態におけると同様である。この第2の実施形態において、第1の実施形態と同一部分には同一符号を付し、その説明を省略する。

【0092】

図5は本発明に係る認証方法の第2の実施形態を説明する図である。この第2の実施形態では、まず、ステップP1において、クライアント20が、 $SIGNAL_{n,1}$ (第一のワンタイムID) を生成するとともに、 ID_c (クライアントID)、 ID_s (サーバID)、DH公開値 g^{xn} および $SIGNAL_{n,1}$ を共有鍵 K_{n-1} (第一の可変共有鍵) で暗号化し、この暗号化データと $SIGNAL_{n,1}$ とをサーバ10に対して送信する処理を実行する。

【0093】

ここで、 $SIGNAL$ は、 i 番目のセッションにおけるクライアント20の j 番目の通信で利用する $SIGNAL$ を $SIGNAL_{i,j}$ 、 i 番目のセッションにおけるサーバ10の j 番目の通信で利用する $SIGNAL$ を $SIGNAL'_{i,j}$

とした場合、次のように生成される。

【0094】

【数3】

$$\text{SIGNAL}_{1,j} = \text{hash}(R, j) \quad i = 1$$

$$\text{SIGNAL}_{i,j} = \text{hash}(K_{i-1}, j) \quad i \geq 2$$

$$\text{SIGNAL}'_{1,j} = \text{hash}'(R, j) \quad i = 1$$

$$\text{SIGNAL}'_{i,j} = \text{hash}'(K_{i-1}, j) \quad i \geq 2$$

【0095】

上記SIGNALの定義式において、hashとhash'は互いに異なるハッシュ関数、RはID発行管理サーバ30からサーバ10とクライアント20の双方に発行されて両者間で共有化された乱数、 K_i はi番目のセッションで共有したDH共通鍵 $g^{x_i y_i}$ （共有鍵）である。

【0096】

すなわち、上記SIGNALの定義式によれば、前回のセッションで生成された共有鍵 K_{i-1} と今回のセッションにおける通信順序jとを引数とするハッシュ関数の関数値を求め、この関数値を今回のセッションのj番目の通信に用いるSIGNALとしている。ただし、最初のセッション（ $i=1$ ）では、サーバ・クライアント間で予め共有化された乱数Rと当該セッションにおける通信順序jとを引数とするハッシュ関数の関数値を求め、この関数値を最初のセッションのj番目の通信に用いるSIGNALとしている。

【0097】

次いで、ステップP2では、サーバ10が、 $\text{SIGNAL}_{n,1}$ を演算により求め、この演算結果と、クライアント20から受信した $\text{SIGNAL}_{n,1}$ との照合により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 K_{i-1} を用いて暗号化データを復号し、この復号したデータに含まれる、IDc、IDsおよび $\text{SIGNAL}_{n,1}$ に基づいて、クライアント20の正当性を判定する処理を実行する。

【0098】

上記判定の結果、受信データと、サーバ10に予め格納された記憶データとが

一致して、クライアント20が正当であると判定される場合には、前述したSIGNALの定義式に従ってSIGNAL'_{n,1}（第二のワンタイムID）を生成するとともに、クライアント20から受信したDH公開値 g^{xn} と当該サーバ10に予め記憶されたDH公開値 g^{yn} とからDH共通鍵 g^{xny_n} を共有鍵 K_n （第二の可変共有鍵）として生成し、この共有鍵 K_n 、IDc、ID sおよびSIGNAL'_{n,1}を引数とするハッシュ関数hの関数値と、DH公開値 g^{yn} と、SIGNAL'_{n,1}とをクライアント20に対して送信する処理を実行する（ステップP3）。一方、受信データと記憶データとが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0099】

次いで、ステップP4では、クライアント20が、SIGNAL'_{n,1}を演算により求め、この演算結果と、サーバ10から受信したSIGNAL'_{n,1}との照合により、サーバ10を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ10から受信したDH公開値 g^{yn} と当該クライアント20に予め記憶されたDH公開値 g^{xn} とからDH共通鍵 g^{xny_n} を共有鍵 K_n として生成するとともに、この共有鍵 K_n を用いてハッシュ関数hの関数値を演算により求め、この演算結果と、サーバ10から受信したハッシュ関数hの関数値との照合により、サーバ10の正当性を判定する処理を実行する。

【0100】

上記判定の結果、受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

【0101】

なお、クライアント20が共有鍵 K_i を共有したことをサーバ10側で確認する必要がある場合には、このステップP4でクライアント20がサーバ10の正当性を判定した後に、共有鍵 K_n 、IDc、ID sを引数とするハッシュ関数h

の関数値をサーバ10に対して送信するようにすればよい。

【0102】

以上のように、この第2の実施形態によれば、前回のセッションで生成された共有鍵 K_{i-1} （可変共有鍵）と今回のセッションにおける通信順序 j とを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションの j 番目の通信にのみ有効なワнтаイムID（ $SIGNAL$ ）として用いるようにしたので、例えば、 n 番目のセッションで生成した共有鍵 K_n が第三者に漏れたとしても、セッション毎に共有鍵 K_n が変化することとなるので、漏れた共有鍵 K_n を用いて生成されたワнтаイムID（ $SIGNAL_{n+1,j}$ 、 $SIGNAL'_{n+1,j}$ ）以外のワнтаイムIDを予測できなくなる。すなわち、盗聴が困難で安全性に優れたワнтаイムIDを生成することが可能になり、ワнтаイムIDの将来にわたる安全性（ PFS ）を実現することが可能になる。

【0103】

また、上記ワнтаイムID（ $SIGNAL$ ）を用いて、クライアント・サーバ間における認証を行うようにしたので、前述した第1の実施形態と同様、大量の計算要求・応答要求などによる計算量やメモリへのDOS攻撃を防止することができ、オープンなネットワーク環境下においても、ID情報の保護を図り、通信の安全性を向上させることができる。

【0104】

なお、DOS攻撃を防止する手法の一つとして、クッキー（乱数）を用いた手法が一般に知られている。この方法によれば、IPアドレスとクッキー生成者しか知らない秘密を組み合わせることにより、同一IPアドレスからのDOS攻撃を防ぐことができる。これに対して、本実施形態の $SIGNAL$ の場合には、DH共通鍵を知らない限り、次回有効となる $SIGNAL$ を予測することができない。よって、毎回の通信に $SIGNAL$ を利用することにより、クッキーと同様の効果が得られる。さらに、クッキーの場合はセッション中にIPアドレスが変わることを許さないが、 $SIGNAL$ は変わっても良い。また、クッキーを用いた場合IPアドレスを偽造したDOS攻撃を防ぐことができないが、ワнтаイムIDではIPアドレスが関係ないためこのような攻撃も防ぐことができる。

【0105】

また、本実施形態において、例えば、クライアント20がプロトコルの最初のメッセージを送り（ステップP1）、サーバ10がそれに対応してDH鍵交換の計算を行い（ステップP2）、2番目のメッセージを送った（ステップP3）場合を考える。もし、サーバ10のメッセージが途中で消失、もしくは攻撃者に横取りされ、クライアント20が受け取ることができなかった場合、クライアント20はもう一度最初のメッセージを送信する必要がある。このとき、サーバ10は正しいクライアント20が通信を送りなおしてきたのか、攻撃者が最初のメッセージを読み取りリプレイ攻撃を行っているのか、判断することができない。そこで、クライアント20はもう一度最初のメッセージを送りなおす場合、最初のチャレンジの際に送ったメッセージと同一の内容のものを送ることにし、サーバ10も以前返信したメッセージのコピーをそのまま送ることにする。これにより、無駄なDH鍵交換の計算を避けることができ、リプレイ攻撃によるDoS攻撃を防ぐことができる。

【0106】

なお、この実施形態では、前回のセッションで生成された共有鍵（DH共通鍵） K_{i-1} と今回のセッションにおける通信順序 j とを引数とするハッシュ関数の関数値を求め、この関数値を当該セッションの j 番目の通信にのみ有効なワンタイムID（SIGNAL）として生成するようにしたが、例えば、次のようにSIGNALを生成することも可能である。

【0107】

【数4】

$$SS_j = h1(K_{i-1})$$

$$SIGNAL_{i,j} = hash(SS_i, j)$$

$$SIGNAL'_{i,j} = hash'(SS_i, j)$$

【0108】

上記SIGNALの定義式において、 SS_i は $(i-1)$ 番目のセッションで共有したDH共通鍵 K_{i-1} を引数とするハッシュ関数の関数値である。

また、この場合には、 i 番目のセッションで用いられる認証用鍵を AK_i 、暗

号化鍵を SK_i として、これら鍵を、例えば、 $AK_i = h_2(K_{i-1})$ 、 $SK_i = h_3(K_{i-1})$ という式から求めるようにしてもよい。なお、 h_1 、 h_2 、 h_3 は、衝突のない一方向性ハッシュ関数である。

【0109】

このように SS_i から認証用鍵および暗号化鍵を生成する場合には、前述したステップP1において、クライアント20が、 ID_c 、 ID_s 、DH公開値 g^{xn} および $SIGNAL_{n,1}$ を暗号化してサーバ10に対して送信する際に、認証用鍵 AK_n を用いるようにする。また、ステップP3において、サーバ10がクライアント20に対して送信するハッシュ関数 h に、暗号化鍵 SK_n 、 ID_c 、 ID_s および $SIGNAL'_{n,1}$ を引数とするハッシュ関数を用いるようにする。

【0110】

そうすることで、攻撃者が、仮に、 SS_i 、 AK_i 、 SK_i の何れか1つの値を知ることができたとしても、その他の値を計算することはできない。よって、攻撃者が i 番目のセッションにおいて正規ユーザに成りすまし、鍵交換を行うためには、 AK_i 、 $SIGNAL$ 、正規ユーザのID情報(ID_s 、 ID_c)が必要となり、暗号通信するためには、 SK_i 、 $SIGNAL$ 、正規ユーザのID情報、通信回数の情報が必要となる。

また、 n 番目のセッションにおけるクライアント20のDH公開値 g^{xn} は、認証鍵 AK_i ($h_2(K_{i-1})$)を用いて暗号化される。よって、 AK_i を知らない攻撃者は g^{xn} を知ることができない。そのため、本方式で生成・共有されるDiffie-Hellman共通鍵は計算量的、かつ情報論的に安全である。

【0111】

[第3の実施形態]

前述した第1および第2の実施形態では、認証と同時にDiffie-Hellman鍵交換を行うようにしたが、この第3の実施形態では、Diffie-Hellman鍵交換を省略するようにしている。この第3の実施形態特有の部分以外は、第1の実施形態におけると同様である。この第3の実施形態において、第1の実施形態と同一部分には同一符号を付し、その説明を省略する。

【0112】

図6は本発明に係る認証方法の第3の実施形態を説明する図である。この第3の実施形態では、まず、クライアント20が、乱数 R_c （第一の乱数）を生成するとともに、サーバ10との間で予め共有化された共有鍵 K_1 （第一の共有鍵）および乱数 R_0 （初期乱数）を引数とする疑似乱数関数 $\text{prf}(K_1, R_0)$ の関数値を SIGNAL_{c1} （第一のワンタイムID）として求め（ステップS11）、この SIGNAL_{c1} と、共有鍵 K_1 で暗号化した乱数 R_c とをサーバ10に対して送信する処理を実行する（ステップS12）。

【0113】

次いで、サーバ10が、乱数 R_s （第二の乱数）を生成するとともに、共有鍵 K_1 で復号した乱数 R_c と共有鍵 K_1 とを引数とする疑似乱数関数 $\text{prf}(K_1, R_c)$ の関数値を SIGNAL_{s1} （第二のワンタイムID）として求め（ステップS13）、この SIGNAL_{s1} と、共有鍵 K_1 で暗号化した乱数 R_s と、乱数 $R_0 + R_c$ （乱数 R_0 、 R_c を引数とする所定の演算結果、例えば、両者の排他的論理和など）とをクライアント20に対して送信する処理を実行する（ステップS14）。

【0114】

次いで、クライアント20が、乱数 R_c と共有鍵 K_1 に基づいて SIGNAL_{s1} を演算により求め、この演算結果とサーバ10から受信した SIGNAL_{s1} との比較により、サーバ10を識別するとともに、乱数 $R_0 + R_c$ の受信データと演算結果との比較により、サーバ10の正当性を判定する処理を実行する（ステップS15）。

【0115】

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、クライアント20が、乱数 R_c および乱数 R_s に基づいて共有鍵 K_2 （第二の共有鍵）を生成するとともに、この共有鍵 K_2 、乱数 R_s および乱数 R_c を引数とする疑似乱数関数 $\text{prf}(K_2, R_s, R_c)$ の関数値を SIGNAL_{c2} （第三のワンタイムID）として求め、この SIGNAL_{c2} と、乱数 $R_c + R_s$ （乱数 R_c 、 R_s を引数とする所定の演算結果）とをサーバ10に対して送信する処理を実行する（ステップS16）。一方、受信デ

ータと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

【0116】

サーバ10は、クライアント20から $SIGNAL_{c2}$ を受信すると、乱数 R_c および乱数 R_s に基づいて共有鍵 K_2 を生成するとともに、共有鍵 K_2 、乱数 R_s および乱数 R_c に基づいて $SIGNAL_{c2}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{c2}$ との比較により、クライアント20を識別するとともに、乱数 $R_c + R_s$ の受信データと演算結果との比較により、クライアント20の正当性を判定する処理を実行する（ステップS17）。

【0117】

上記判定の結果、各々の受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0118】

以上のように、この第3の実施形態によれば、相互認証の過程で生成された乱数と、相互認証の過程で変化する共有鍵 K とを引数とする疑似乱数関数 prf の関数値をワンタイムIDとして用いるようにしたので、前述した第1の実施形態と同様、ワンタイムIDの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

【0119】

〔第4の実施形態〕

前述した第3の実施形態では、ワンタイムID（ $SIGNAL$ ）の生成に用いる共有鍵を相互認証の過程で変化させるようにしたが、この第4の実施形態では、上記共有鍵を固定するようにしている。

【0120】

すなわち、この第4の実施形態では、図7に示すように、先ず、クライアント20が、乱数 R_c （第一の乱数）を生成するとともに、サーバ10との間で予め

共有化された共有鍵 K および乱数 R_0 （初期乱数）を引数とする疑似乱数関数 $\text{prf}(K, R_0)$ の関数値を SIGNAL_{c1} （第一のワンタイムID）として求め（ステップS21）、この SIGNAL_{c1} と、共有鍵 K で暗号化した乱数 R_c とをサーバ10に対して送信する処理を実行する（ステップS22）。

【0121】

次いで、サーバ10が、乱数 R_s （第二の乱数）を生成するとともに、共有鍵 K で復号した乱数 R_c と共有鍵 K とを引数とする疑似乱数関数 $\text{prf}(K, R_c)$ の関数値を SIGNAL_{s1} （第二のワンタイムID）として求め（ステップS23）、この SIGNAL_{s1} と、共有鍵 K で暗号化した乱数 R_s と、乱数 $R_0 + R_c$ （乱数 R_0 、 R_c を引数とする所定の演算結果）とをクライアント20に対して送信する処理を実行する（ステップS24）。

【0122】

次いで、クライアント20が、乱数 R_c および共有鍵 K に基づいて SIGNAL_{s1} を演算により求め、この演算結果とサーバ10から受信した SIGNAL_{s1} との比較により、サーバ10を識別するとともに、乱数 $R_0 + R_c$ の受信データと演算結果との比較により、サーバ10の正当性を判定する処理を実行する（ステップS25）。

【0123】

上記判定の結果、各々の受信データと演算結果とが一致して、サーバ10が正当であると判定される場合には、クライアント20が、乱数 R_c 、乱数 R_s および共有鍵 K を引数とする疑似乱数関数 $\text{prf}(K, R_s, R_c)$ の関数値を SIGNAL_{c2} （第三のワンタイムID）として求め、この SIGNAL_{c2} と、乱数 $R_c + R_s$ （乱数 R_c 、 R_s を引数とする所定の演算結果）とをサーバ10に対して送信する処理を実行する（ステップS26）。一方、受信データと演算結果とが一致せず、サーバ10が正当でないと判定される場合には、サーバ10へのアクセスを中止して、当該認証処理を終了する。

【0124】

サーバ10は、クライアント20から SIGNAL_{c2} を受信すると、乱数 R_c 、乱数 R_s および共有鍵 K に基づいて SIGNAL_{c2} を演算により求め、この演

算結果とクライアント20から受信した $SIGNAL_{c2}$ との比較により、クライアント20を識別するとともに、乱数 $R_c + R_s$ の受信データと演算結果との比較により、クライアント20の正当性を判定する処理を実行する（ステップS27）。

【0125】

上記判定の結果、各々の受信データと演算結果とが一致して、クライアント20が正当であると判定される場合には、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと演算結果とが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0126】

以上のように、この第4の実施形態によれば、相互認証の過程で生成された乱数と共有鍵 K とを引数とする疑似乱数関数 prf の関数値をワンタイムIDとして用いるようにしたため、例えば、共有鍵 K が第三者に漏れたとしても、乱数によって疑似乱数関数 prf の関数値が相互認証の過程で順次変化することとなるので、相互認証の過程で生成される乱数がわからない限り、ワンタイムIDを予測することができない。したがって、前述した第1～第3の実施形態と同様、ワンタイムIDの安全性を高めることができ、迅速かつ安全な相互認証を実現することができる。

【0127】

〔第5の実施形態〕

図8は本発明に係る認証方法の第5の実施形態を説明する図である。この第5の実施形態では、先ず、クライアント20が、乱数 R_{ci} （第一の乱数）を生成するとともに、サーバ10との間で予め共有化された共有鍵 K_i 、乱数 R_{ci-1} （第一の記憶乱数）および乱数 R_{si-1} （第二の記憶乱数）を引数とする疑似乱数関数 $prf(K_i, R_{ci-1}, R_{si-1})$ の関数値を $SIGNAL_{ci}$ （第一のワンタイムID）として求める処理を実行する（ステップS31）。

【0128】

なお、 R_{ci} は i 番目のセッションでクライアント20により生成された乱数、

R_{si} は i 番目のセッションでサーバ 10 により生成された乱数、 K_i は i 番目のセッションで使用する可変共有鍵をそれぞれ示している。また、前回 ($i-1$ 番目) のセッションで生成された乱数 R_{ci-1} 、 R_{si-1} は、サーバ 10 とクライアント 20 の各記憶装置 13、23 の記憶領域に格納されており、これら乱数 R_{ci-1} 、 R_{si-1} に基づいて、共有鍵 K_i が生成されるようになっている。

【0129】

そして、クライアント 20 は、 $SIGNAL_{ci}$ を生成した後、 ID_c (クライアント ID)、 ID_s (サーバ ID) および乱数 R_{ci} を共有鍵 K_i で暗号化した暗号化データ $E_{Ki}(ID_c, ID_s, R_{ci})$ と、 $SIGNAL_{ci}$ とをサーバ 10 に対して送信する処理を実行する (ステップ S32)。

【0130】

サーバ 10 は、クライアント 20 から $SIGNAL_{ci}$ を受信すると、共有鍵 K_i 、乱数 R_{ci-1} および乱数 R_{si-1} に基づいて $SIGNAL_{ci}$ を演算により求め、この演算結果とクライアント 20 から受信した $SIGNAL_{ci}$ との比較により、クライアント 20 を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 K_i を用いて暗号化データ $E_{Ki}(ID_c, ID_s, R_{ci})$ を復号し、この復号したデータに含まれる ID_c および ID_s に基づいて、クライアント 20 の正当性を判定する処理を実行する。

【0131】

上記判定の結果、受信データと、サーバ 10 に予め格納された記憶データとが一致して、クライアント 20 が正当であると判定される場合には、乱数 R_{si} (第二の乱数) を生成するとともに、乱数 R_{ci} 、乱数 R_{si-1} および共有鍵 K_i を引数とする疑似乱数関数 $prf(K_i, R_{ci}, R_{si-1})$ の関数値を $SIGNAL_{si}$ (第二のワンタイム ID) として求める。そして、乱数 R_{ci-1} 、 R_{si-1} を格納していた記憶領域に、乱数 R_{ci} 、 R_{si} をそれぞれ格納するとともに、これら乱数 R_{ci} 、 R_{si} に基づき共有鍵 K_{i+1} を生成して記憶する処理を実行する (ステップ S33)。

【0132】

次いで、サーバ 10 は、 ID_c 、 ID_s および乱数 R_{si} を共有鍵 K_i で暗号化

した暗号化データ $E_{K_i}(ID_s, ID_c, R_{si})$ と、 $SIGNAL_{si}$ とをクライアント 20 に対して送信する処理を実行する（ステップ S34）。

一方、受信データと記憶データとが一致せず、クライアント 20 が正当でないと判定される場合には、クライアント 20 からのアクセスを拒否して、当該認証処理を終了する。

【0133】

クライアント 20 は、サーバ 10 から $SIGNAL_{si}$ を受信すると、共有鍵 K_i 、乱数 R_{ci} および乱数 R_{si-1} に基づいて $SIGNAL_{si}$ を演算により求め、この演算結果とクライアント 20 から受信した $SIGNAL_{si}$ との比較により、サーバ 10 を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合には、共有鍵 K_i を用いて暗号化データ $E_{K_i}(ID_s, ID_c, R_{si})$ を復号し、この復号したデータに含まれる ID_c および ID_s に基づいて、サーバ 10 の正当性を判定する処理を実行する。サーバ 10 を識別できる場合、通信相手を特定できるだけでなく、サーバ 10 が乱数 R_{ci} を受け取ったことも確認することができる。

【0134】

そして、上記判定の結果、受信データと、クライアント 20 に予め格納された記憶データとが一致して、サーバ 10 が正当であると判定される場合には、乱数 R_{ci-1} 、 R_{si-1} を格納していた記憶領域に、乱数 R_{ci} 、 R_{si} をそれぞれ格納して、これら乱数 R_{ci} 、 R_{si} に基づき共有鍵 K_{i+1} を生成・記憶した後（ステップ S35）、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ 10 が正当でないと判定される場合には、サーバ 10 からのアクセスを拒否して、当該認証処理を終了する。

【0135】

以上のように、この第 5 の実施形態によれば、前述した第 3 の実施形態と同様の作用・効果が得られるのに加えて、 ID_c 、 ID_s および乱数 R_{si} を共有鍵 K_i で暗号化した暗号化データ $E_{K_i}(ID_s, ID_c, R_{si})$ を通信相手に送信するようにしたことにより、例えば、攻撃者によって暗号化データが書き換えられた場合においても、暗号化データに含まれる ID 情報（ ID_s 、 ID_c ）が正し

く復号されないために、このデータを受け取ったサーバ10またはクライアント20は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、 $SIGNAL_{ci}$ の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるID情報(IDs、IDc)を参照することによって、通信相手を容易に特定することができる。

【0136】

さらに、この第5の実施形態によれば、通信相手がサーバ・クライアントのID情報(IDs、IDc)を正しく暗号化できているか否かを確認することによって、通信相手の正当性を判定するようにしたので、前述した第3の実施形態では3回必要であった通信回数を2回に低減することが可能となり、より効率的な認証が可能となる。

【0137】

[第6の実施形態]

図9は本発明に係る認証方法の第6の実施形態を説明する図である。この第6の実施形態では、まず、クライアント20が、乱数 R_{ci} （第一の乱数）を生成するとともに、サーバ10との間で予め共有化された固定共有鍵 K 、乱数 R_{ci-1} （第一の記憶乱数）および乱数 R_{si-1} （第二の記憶乱数）を引数とする疑似乱数関数 $prf(K, R_{ci-1}, R_{si-1})$ の関数値を $SIGNAL_{ci}$ （第一のワンタイムID）として求める処理を実行する（ステップS41）。

【0138】

なお、 R_{ci} は i 番目のセッションでクライアント20により生成された乱数、 R_{si} は i 番目のセッションでサーバ10により生成された乱数をそれぞれ示している。また、前回($i-1$ 番目)のセッションで生成された乱数 R_{ci-1} 、 R_{si-1} は、サーバ10とクライアント20の各記憶装置13、23の記憶領域に格納されている。

【0139】

そして、クライアント20は、 $SIGNAL_{ci}$ を生成した後、IDc（クライアントID）、IDs（サーバID）および乱数 R_{ci} を共有鍵 K で暗号化した暗

号化データ $E_K (ID_c, ID_s, R_{ci})$ と、 $SIGNAL_{ci}$ とをサーバ10に対して送信する処理を実行する（ステップS42）。

【0140】

サーバ10は、クライアント20から $SIGNAL_{ci}$ を受信すると、共有鍵 K 、乱数 R_{ci-1} および乱数 R_{si-1} に基づいて $SIGNAL_{ci}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{ci}$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、共有鍵 K を用いて暗号化データ $E_K (ID_c, ID_s, R_{ci})$ を復号し、この復号したデータに含まれる ID_c および ID_s に基づいて、クライアント20の正当性を判定する処理を実行する。

【0141】

上記判定の結果、受信データと、サーバ10に予め格納された記憶データとが一致して、クライアント20が正当であると判定される場合には、乱数 R_{si} （第二の乱数）を生成するとともに、乱数 R_{ci} 、乱数 R_{si-1} および共有鍵 K を引数とする疑似乱数関数 $prf(K, R_{ci}, R_{si-1})$ の関数値を $SIGNAL_{si}$ （第二のワンタイムID）として求める。そして、乱数 R_{ci-1} 、 R_{si-1} を格納していた記憶領域に、乱数 R_{ci} 、 R_{si} をそれぞれ格納する処理を実行する（ステップS43）。

【0142】

次いで、サーバ10は、 ID_c 、 ID_s および乱数 R_{si} を共有鍵 K で暗号化した暗号化データ $E_K (ID_s, ID_c, R_{si})$ と、 $SIGNAL_{si}$ とをクライアント20に対して送信する処理を実行する（ステップS44）。

一方、受信データと記憶データとが一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0143】

クライアント20は、サーバ10から $SIGNAL_{si}$ を受信すると、共有鍵 K 、乱数 R_{ci} および乱数 R_{si-1} に基づいて $SIGNAL_{si}$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_{si}$ との比較により、サー

サーバ10を識別し、識別できない場合には、通信を拒否する。一方、識別できる場合には、共有鍵 K を用いて暗号化データ $E_K(ID_s, ID_c, R_{si})$ を復号し、この復号したデータに含まれる ID_c および ID_s に基づいて、サーバ10の正当性を判定する処理を実行する。サーバ10を識別できる場合、通信相手を特定できるだけでなく、サーバ10が乱数 R_{ci} を受け取ったことも確認することができる。

【0144】

そして、上記判定の結果、受信データと、クライアント20に予め格納された記憶データとが一致して、サーバ10が正当であると判定される場合には、乱数 R_{ci-1} 、 R_{si-1} を格納していた記憶領域に、乱数 R_{ci} 、 R_{si} をそれぞれ格納して、これら乱数 R_{ci} 、 R_{si} に基づき共有鍵 K を生成・記憶した後（ステップS45）、当該認証処理を終了して、次のデータ伝送処理に移行する。一方、受信データと記憶データとが一致せず、サーバ10が正当でないと判定される場合には、サーバ10からのアクセスを拒否して、当該認証処理を終了する。

【0145】

以上のように、この第6の実施形態によれば、前述した第4の実施形態と同様の作用・効果が得られるのに加えて、例えば、攻撃者によって暗号化データが書き換えられた場合においても、データを受け取ったサーバ10またはクライアント20は、送られてきた暗号化データが誤ったものであることを容易に検出でき、乱数を受け取らずに廃棄することが可能となる。また、 $SIGNAL_{ci}$ の値が他の複数のクライアントと重複した場合においても、暗号化データに含まれるID情報（ ID_s 、 ID_c ）を参照することによって、通信相手を容易に特定することができる。さらに、この第6の実施形態によれば、前述した第4の実施形態では3回必要であった通信回数を2回に低減することが可能となり、より効率的な認証が可能となる。

【0146】

〔第7の実施形態〕

この第7の実施形態では、ワンタイムIDを用いたリプレイ攻撃の防止方法について説明する。リプレイ攻撃とは、過去に正式な通信者が送信したときに有効

であった通信情報を攻撃者（第三者）が盗聴し、再利用する攻撃のことである。

【0147】

先ず、OSP A (Optimal Strong Password Authentication) と呼ばれるパスワードを利用した認証方式 (Chun-Li LIN, Hung-Min SUN, Tzonelih HWANG, Attacks and Solutions on Strong- Password Authentication, IEICE TRANS. COMMUN., VOL.E84-B, NO.9, September 2001.) について、図10に基づいて説明する。

当該認証に先立って、クライアント20には、ハッシュ関数 h およびパスワード P が予め記憶保持されており、サーバ10には、ハッシュ関数 h 、セッション回数 n 、IDc (クライアントID) および検証用情報 $h^2 (P @ n)$ が予め記憶保持されている。検証用情報 $h^2 (P @ n)$ は、クライアント20の正当性を検証するための情報であって、パスワード P と通信回数 n の排他的論理和を用いてハッシュ関数 h により生成された情報である。なお、 $h^2 (P @ n)$ は、ハッシュ関数 h の計算を2回行うこと、つまり $h (h (P @ n))$ を示しており、この数式中の $@$ は排他的論理和を示している。

【0148】

この認証方式では、先ず、クライアント20がサーバ10に対してIDcを送信する (ステップS51)。

サーバ10は、クライアント20からIDcを受信すると、この受信したIDcと、予め記憶しているIDcとの比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ10に対してセッション回数 n を送信する (ステップS52)。

【0149】

クライアント20は、サーバ10からセッション回数 n を受信すると、この受信したセッション回数 n 、予め記憶しているハッシュ関数 h およびパスワード P を用いて、第1～第3の認証用情報 $C1$ 、 $C2$ 、 $C3$ を生成し (ステップS53)、これら $C1$ 、 $C2$ 、 $C3$ をサーバ10に対して送信する (ステップS54)。ここで、 $C1 = h (P @ n) @ h^2 (P @ n)$ 、 $C2 = h^2 (P @ (n+1)) @ h (P @ n)$ 、 $C3 = h^3 (P @ (n+1))$ である。

【0150】

サーバ10は、クライアント20からC1、C2、C3を受信すると、先ず、受信した $C1 \neq C2$ であることを確認する。これは、 $C1 = h(P@n) @ h^2(P@n)$ 、 $C2 = h(P@n) @ h^2(P@n)$ 、 $C3 = h^3(P@n)$ と計算して送った場合においても、サーバ10がクライアント20を認証してしまい、次の検証用情報として、 $h^2(P@(n+1))$ ではなく $h^2(P@n)$ を記憶してしまう不具合が発生する可能性があることから、このような不具合の発生を防ぐために行われるものである。

【0151】

次いで、サーバ10は、C1、C2から、 $h(P@n)$ 、 $h^2(P@(n+1))$ を演算により求める。すなわち、受信したC1と、予め記憶している検証用情報 $h^2(P@n)$ との排他的論理和を求めることで $h(P@n)$ を導き出し、この $h(P@n)$ と受信したC2との排他的論理和を求めることで $h^2(P@(n+1))$ を導き出す。

【0152】

次いで、予め記憶しているハッシュ関数 h を用いて、求めた $h(P@n)$ から $h(h(P@n))$ を計算し、この $h(h(P@n))$ が、予め記憶している検証用情報 $h^2(P@n)$ と一致するか否かを検証する。同時に、求めた $h^2(P@(n+1))$ から上記ハッシュ関数 h を用いて $h(h^2(P@(n+1)))$ を計算し、この $h(h^2(P@(n+1)))$ が、受信したC3と一致するか否かを検証する（ステップS55）。

【0153】

これら検証の結果、何れもが一致して、クライアント20が正当であると判定される場合には、検証用情報を $h^2(P@n)$ から $h^2(P@(n+1))$ に更新し、セッション回数を n から $n+1$ に更新した後、クライアント20からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0154】

上記認証方式によれば、盗聴者に対して安全な認証を行うことができるとともに、セッション毎に検証用情報を $h^2(P@n)$ から $h^2(P@(n+1))$ へと更新することができるという利点がある。

しかしながら、上記認証方式にあっては、一度使用された認証情報 $C1$ 、 $C2$ 、 $C3$ をもう一度利用することによるリプレイ攻撃を防止することができないという問題点があった。

【0155】

そこで、本発明者等は、このような問題点を解決する認証方式として、次のような認証方式を開発した。

図11は、本発明に係る認証方法の第7の実施形態を説明する図である。この図11に示すように、クライアント20に、ハッシュ関数 h およびパスワード P が予め記憶保持され、サーバ10に、ハッシュ関数 h 、セッション回数 n 、 IDc および検証用情報 $h^2(P@n)$ が予め記憶保持されている場合には、先ず、クライアント20がサーバ10に対して IDc を送信する（ステップS61）。

【0156】

サーバ10は、クライアント20から IDc を受信すると、この受信した IDc と、予め記憶している IDc との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、サーバ10に対してセッション回数 n を送信する（ステップS62）。

【0157】

クライアント20は、サーバ10からセッション回数 n を受信すると、この受信したセッション回数 n 、予め記憶しているハッシュ関数 h およびパスワード P を用いて、第1～第3の認証用情報 $C1$ 、 $C2$ 、 $C3$ 、 $SIGNAL_n$ を生成し（ステップS63）、これら $C1$ 、 $C2$ 、 $C3$ 、 $SIGNAL_n$ をサーバ10に対して送信する（ステップS64）。ここで、 $C1 = h(P@n) @ h^2(P@n)$ 、 $C2 = h^2(P@(n+1)) @ h(P@n)$ 、 $C3 = h^3(P@(n+1))$ 、 $SIGNAL_n = h(h^2(P@n), n)$ である。すなわち、 n 番目のセッションで使用するワンタイムIDである $SIGNAL_n$ が、検証用情報 h

² (P@n) およびセッション回数 n を引数とするハッシュ関数 h の関数値となっている。

【0158】

サーバ10は、クライアント20からC1、C2、C3、 $SIGNAL_n$ を受信すると、先ず、予め記憶している検証用情報 $h^2(P@n)$ とセッション回数nとに基づいて $SIGNAL_n$ を演算により求め、この演算結果とクライアント20から受信した $SIGNAL_n$ との比較により、クライアント20を識別し、識別できない場合には、通信を拒否する。識別できる場合には、受信したC1≠C2であることを確認した後、C1およびC2から $h(P@n)$ 、 $h^2(P@n)$ を演算により求める。

【0159】

次いで、サーバ10は、予め記憶しているハッシュ関数hを用いて、求めた $h(P@n)$ から $h(h(P@n))$ を計算し、この $h(h(P@n))$ が、予め記憶している検証用情報 $h^2(P@n)$ と一致するか否かを検証する。同時に、求めた $h^2(P@n)$ から上記ハッシュ関数hを用いて $h(h^2(P@n))$ を計算し、この $h(h^2(P@n))$ が、受信したC3と一致するか否かを検証する(ステップS65)。

【0160】

これら検証の結果、何れもが一致して、クライアント20が正当であると判定される場合には、検証用情報を $h^2(P@n)$ から $h^2(P@n+1)$ に更新し、セッション回数をnからn+1に更新した後、クライアント20からのアクセスを承諾して、当該認証処理を終了する。一方、上記検証の結果、少なくとも何れか一方が一致せず、クライアント20が正当でないと判定される場合には、クライアント20からのアクセスを拒否して、当該認証処理を終了する。

【0161】

上記認証方式によれば、検証用情報である $h^2(P@n)$ が攻撃者に知られる虞がないので、次のセッションの $SIGNAL$ が攻撃者に予測されることはない。しかも、 $SIGNAL$ は他のセッションで使用できないので、攻撃者によるリプレイ攻撃を効果的に防ぐことができる。

【0162】

なお、図12に示すように、ハッシュ関数hおよびパスワードPに加えて、予めセッション回数nもクライアント20に記憶保持されている場合には、前述したステップS61、S62の処理を省略することが可能である。したがって、この場合には、ID情報(IDc)の盗聴に対する保護を図りつつも、攻撃者によるリプレイ攻撃を効果的に防ぐことが可能である。

【0163】

なお、以上の各実施形態においては、複数の装置間の認証にワンタイムIDを用いるようにしたが、一装置内の複数のアプリケーション間の認証にワンタイムIDを用いることも可能である。また、以上の各実施形態においては、本発明に係る認証方法をクライアントサーバシステムに適用した場合について例示したが、これに限られるものではなく、例えば、P2P (Peer to Peer) システムに本発明に係る認証方法を適用することも可能である。

【0164】

また、本発明に係る認証方法をユーザによるアクセス毎に利用することも可能であり、その場合には、ユーザによるパスワードの入力を促して、パスワード、若しくはパスワードから生成した値(ワンタイムパスワードを含む。)をワンタイムIDとともに認証用のデータとして用いることが可能である。

【0165】

【発明の効果】

以上説明したように、請求項1～請求項6の何れかに記載の発明によれば、盗聴が困難で安全性に優れたワンタイムIDを生成することが可能となり、ワンタイムIDの将来にわたる安全性(PFS)を実現することが可能となる。

【0166】

請求項7～請求項35の何れかに記載の発明によれば、請求項1～請求項6の何れかに記載のワンタイムIDの生成方法により生成されたワンタイムIDを用いて、装置間(クライアント・サーバ間)の認証を行うようにしたので、第三者が送信者・受信者を特定できなくなる一方で、正当な送信者・受信者であればワンタイムIDを識別情報として把握できるようになる。

したがって、D o S 攻撃やなりすまし等に対する耐性を強化することができ、オープンなネットワーク環境下においても、I D 情報の保護を図り、通信の安全性を向上させることができる。また、リモートアクセスが可能になり、利便性の向上を図ることもできる。

【 0 1 6 7 】

請求項 8 に記載の発明によれば、従来の鍵交換・認証方式では 3 回必要であった通信回数を 2 回に低減することが可能になり、迅速かつ安全な認証および鍵交換を実現することが可能になる。

【図面の簡単な説明】

【図 1】

本発明に係る認証システムの一実施形態を示す概略構成図である。

【図 2】

図 1 のサーバの概略構成を示すブロック図である。

【図 3】

図 1 のクライアントの概略構成を示すブロック図である。

【図 4】

本発明に係る認証方法の第 1 の実施形態を説明する図である。

【図 5】

本発明に係る認証方法の第 2 の実施形態を説明する図である。

【図 6】

本発明に係る認証方法の第 3 の実施形態を説明する図である。

【図 7】

本発明に係る認証方法の第 4 の実施形態を説明する図である。

【図 8】

本発明に係る認証方法の第 5 の実施形態を説明する図である。

【図 9】

本発明に係る認証方法の第 6 の実施形態を説明する図である。

【図 1 0】

O S P A と呼ばれる従来の認証方法を説明する図である。

【図 1 1】

本発明に係る認証方法の第 7 の実施形態を説明する図である。

【図 1 2】

図 1 1 の変形例を説明する図である。

【図 1 3】

P - S I G M A と呼ばれる従来の認証方法を説明する図である。

【符号の説明】

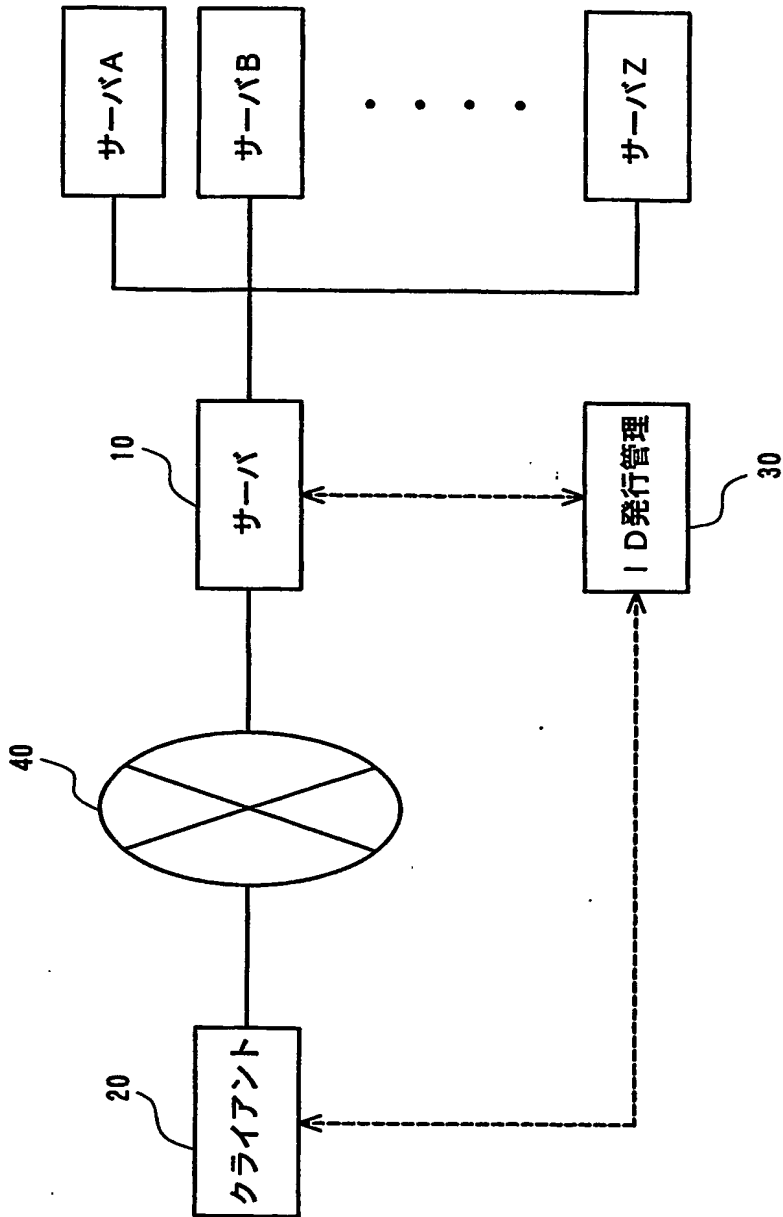
1 0 サーバ（第二装置）

2 0 クライアント（第一装置）

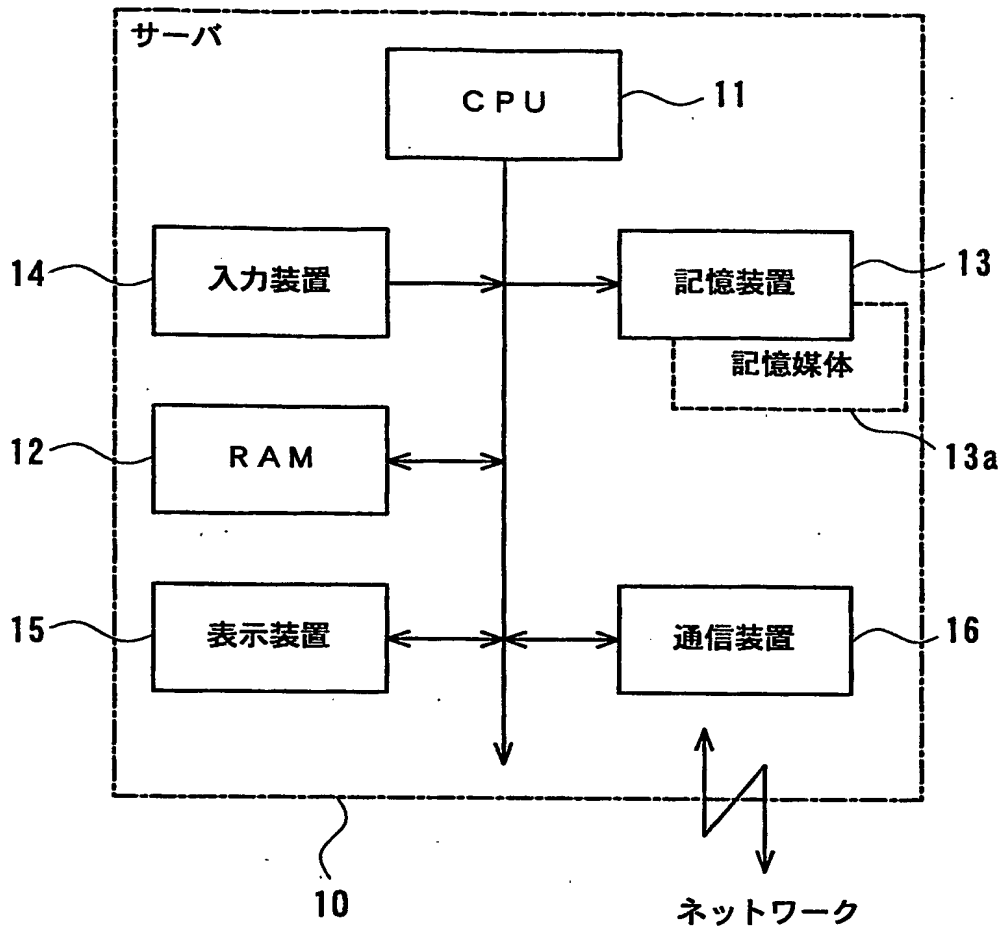
1 1、2 1 CPU

【書類名】 図面

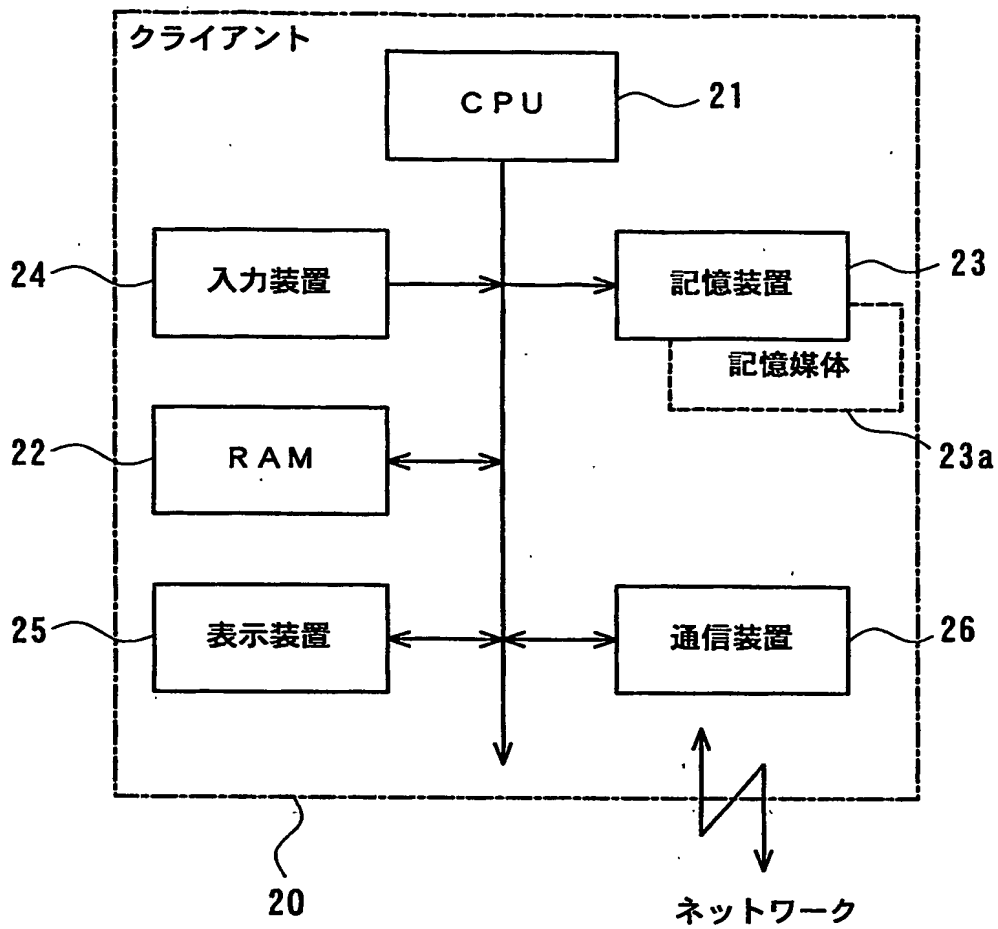
【図 1】



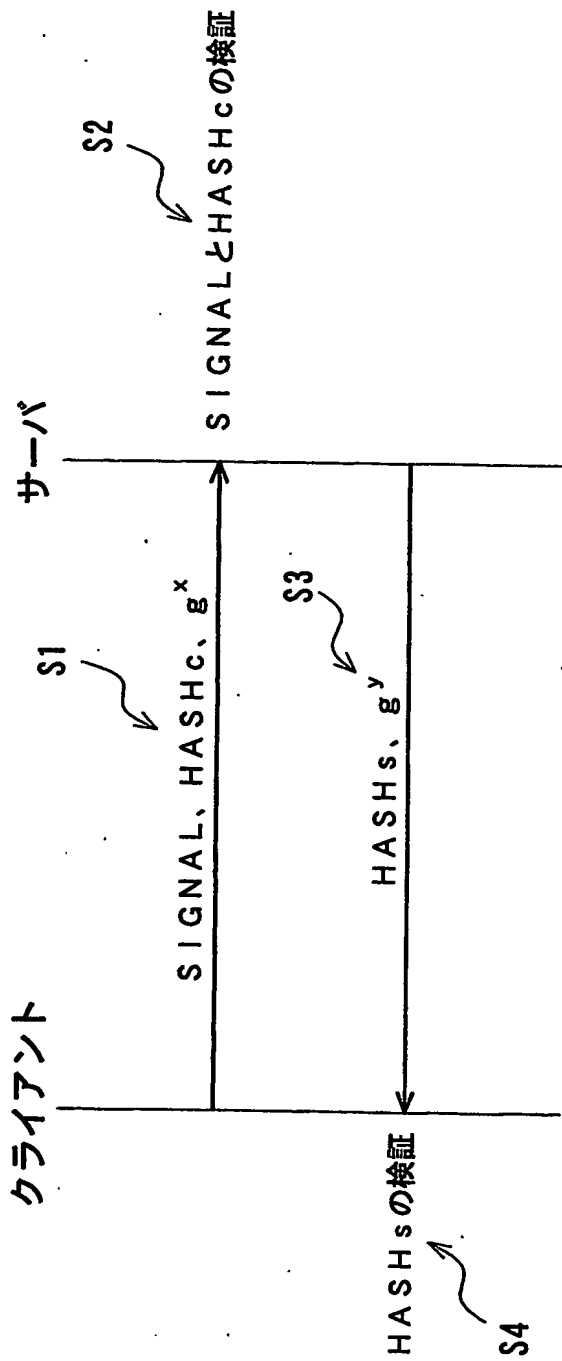
【図2】



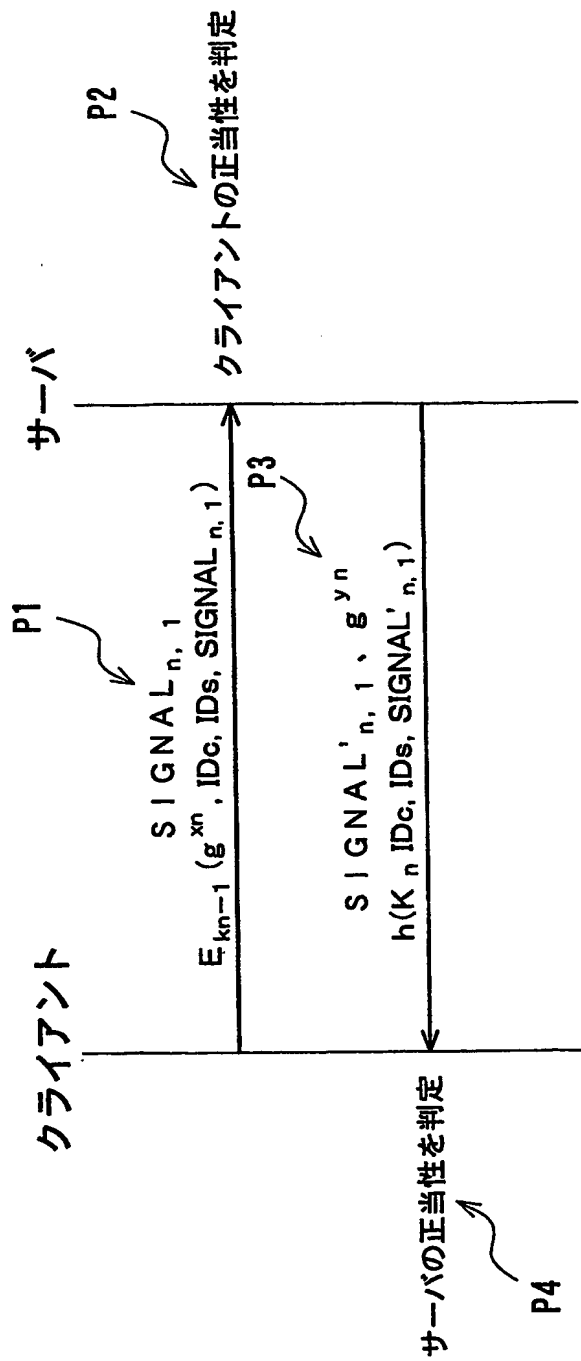
【図3】



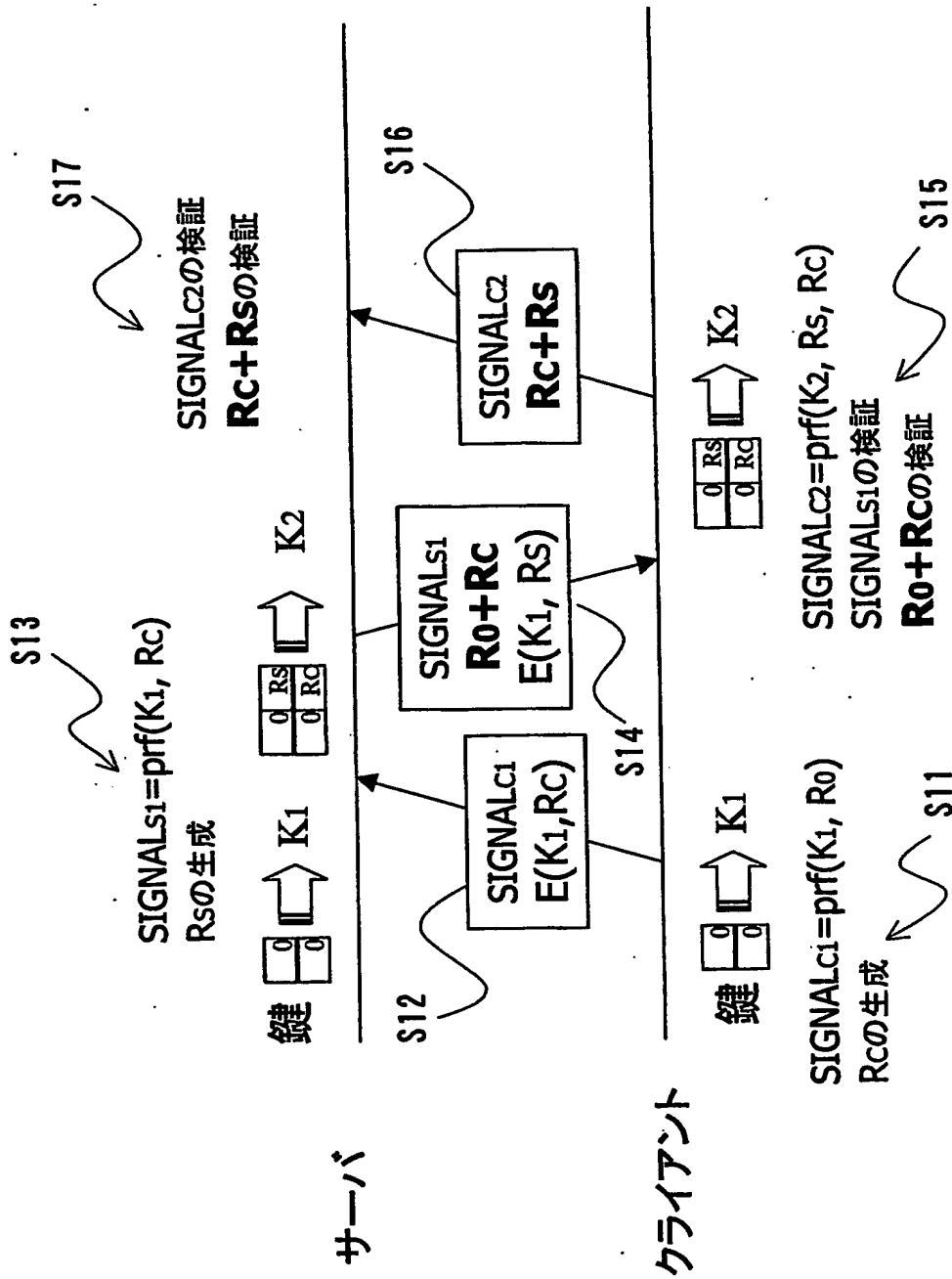
【図4】



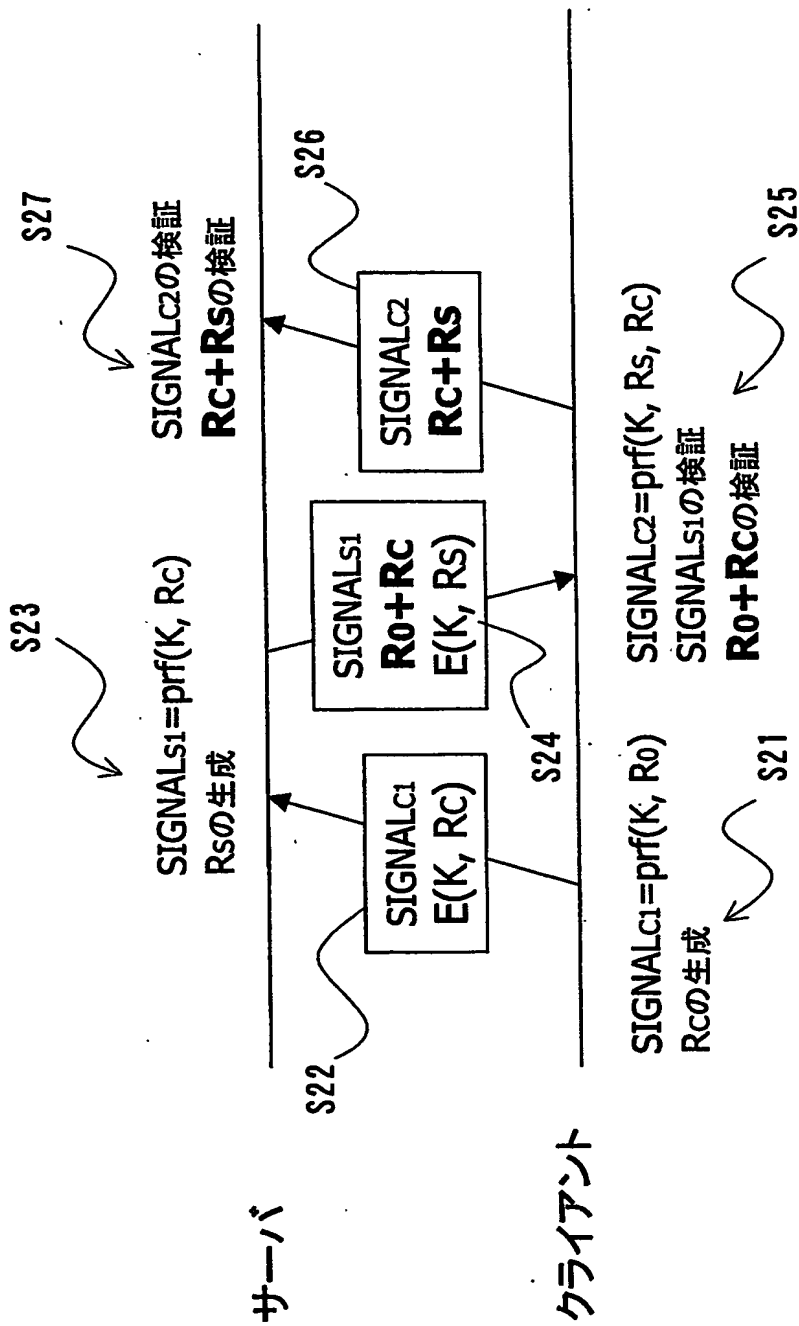
【図5】



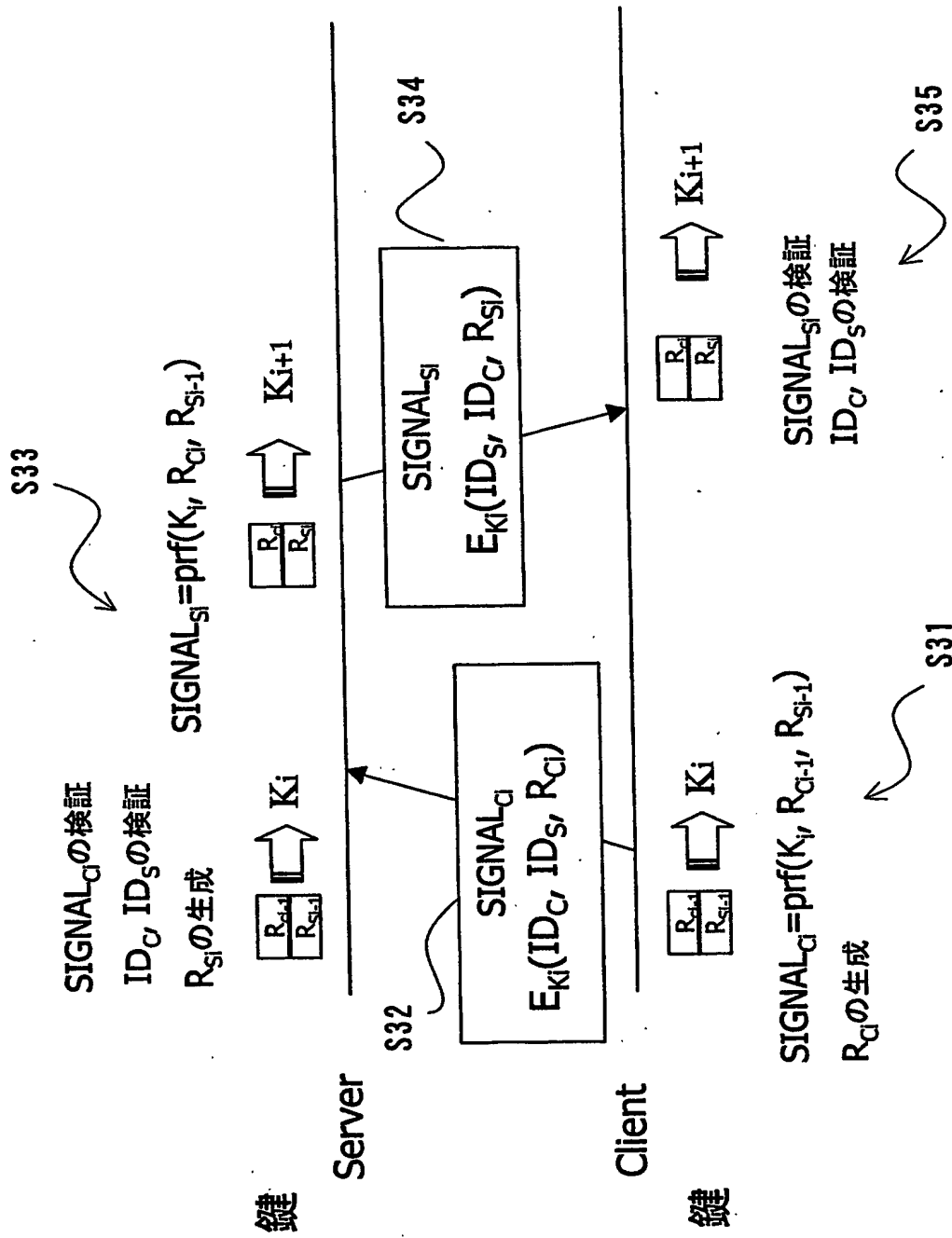
【図6】



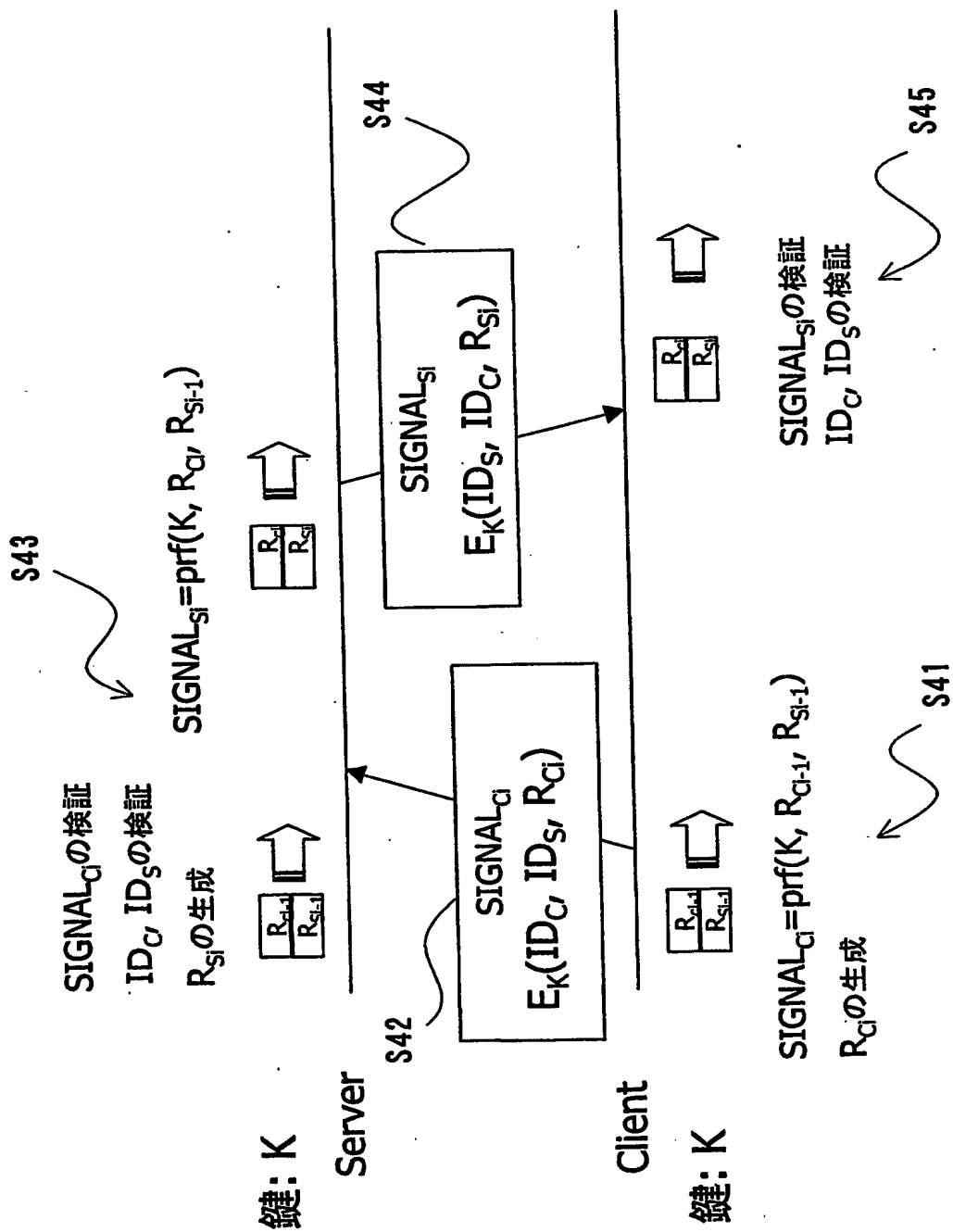
【図7】



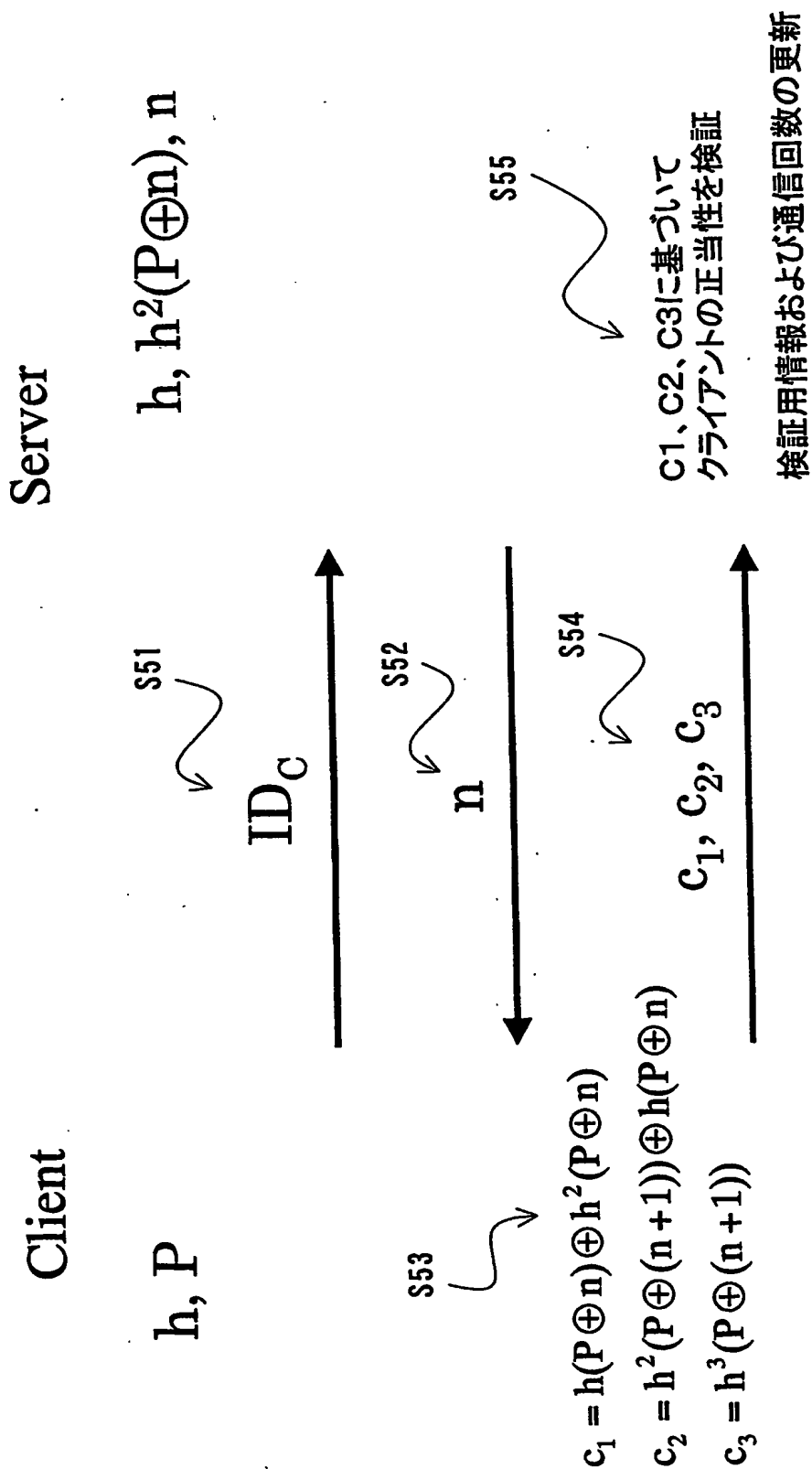
【図 8】



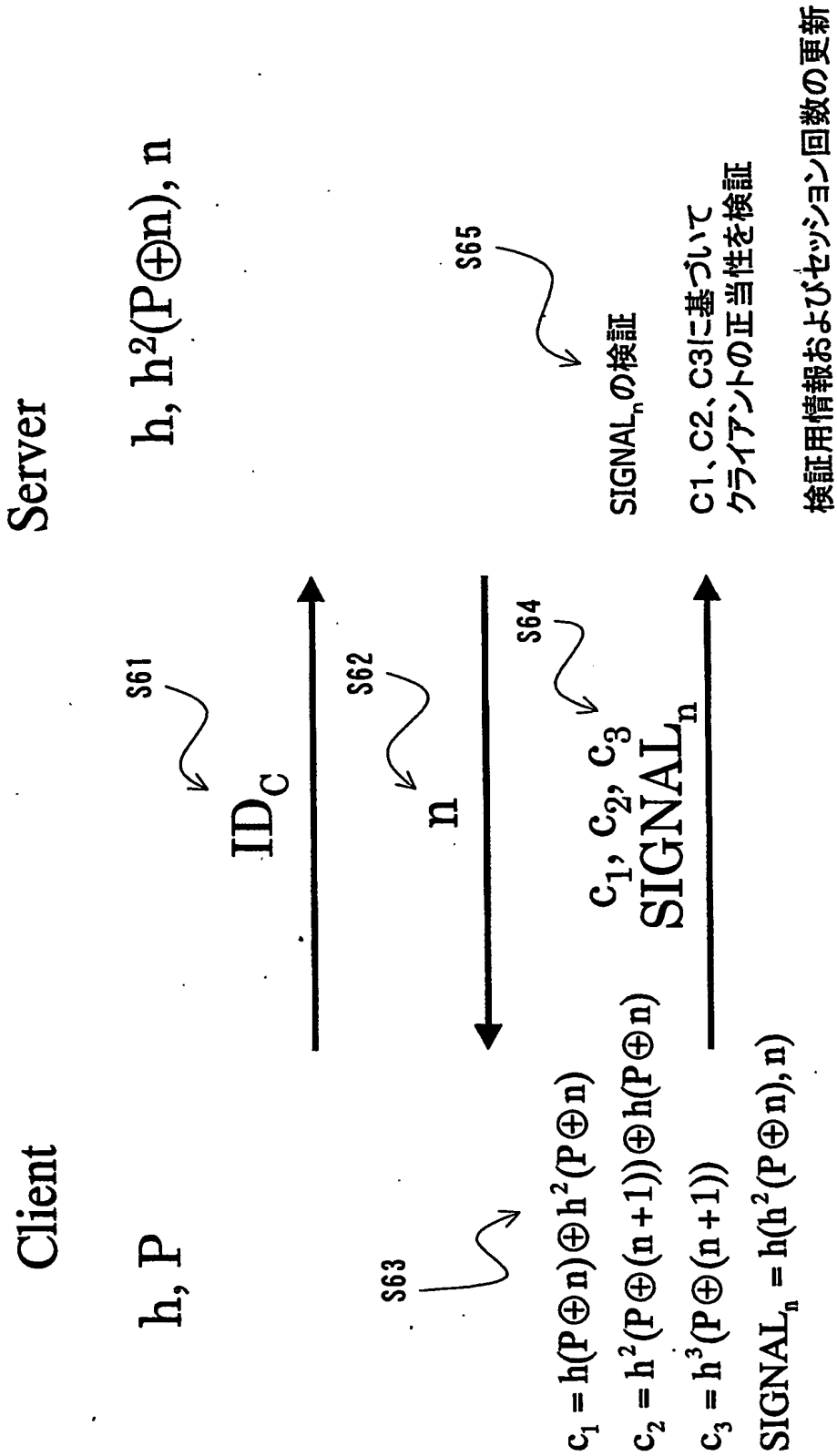
【図9】



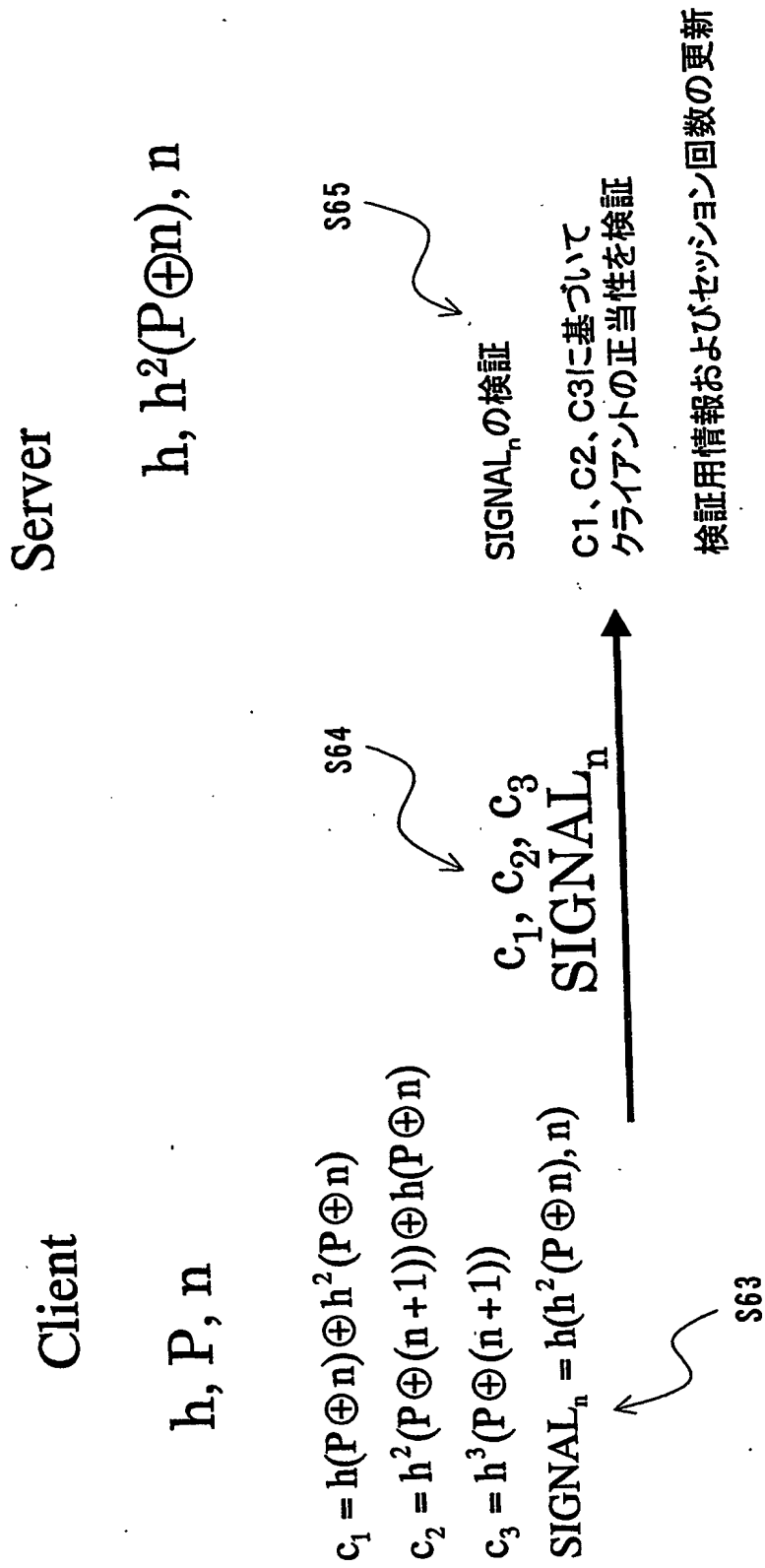
【図 10】



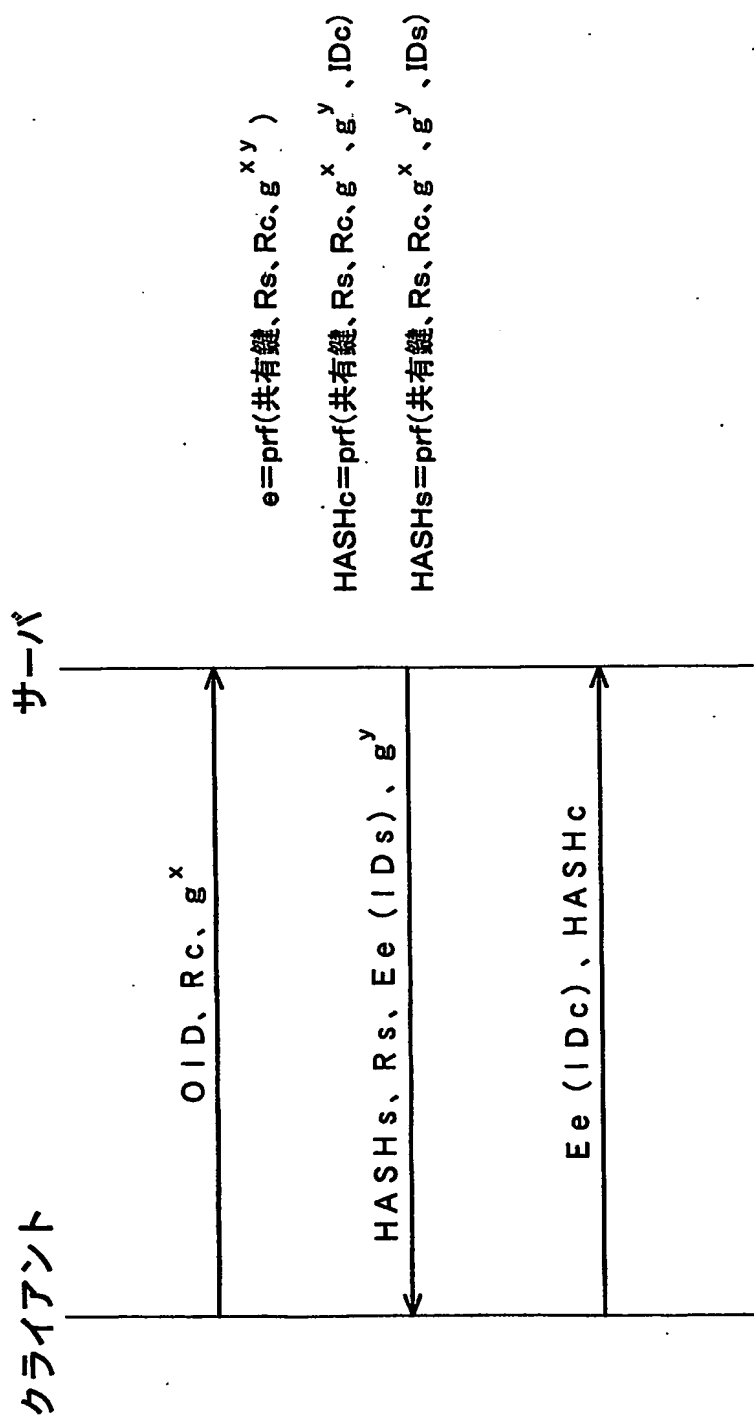
【図 11】



【図 12】



【図 13】



【書類名】 要約書

【要約】

【課題】 盗聴が困難で安全性に優れたワンタイムIDの生成方法、上記ワンタイムIDを用いた認証方法、認証システム、サーバ、クライアントおよびプログラムを提供する。

【解決手段】 複数の装置間またはアプリケーション間の認証において一回限り使用可能な識別情報をワンタイムIDとして、当該ワンタイムIDを生成する方法である。認証を行う装置またはアプリケーションの各々において、認証が必要な所定の通信単位毎に変化する可変共有鍵を生成するとともに、この可変共有鍵を引数とする一方向関数の関数値を求め、この関数値から上記ワンタイムIDを生成するようにした。

【選択図】 図4

認定・付加情報

特許出願の番号	特願2003-069375
受付番号	50300419397
書類名	特許願
担当官	小野寺 光子 1721
作成日	平成15年 3月17日

<認定情報・付加情報>

【提出日】	平成15年 3月14日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [800000035]

1. 変更年月日	2000年10月18日
[変更理由]	住所変更
住 所	福岡県福岡市東区箱崎6丁目10番1号
氏 名	株式会社産学連携機構九州

出 願 人 履 歴 情 報

識別番号

[500196087]

1. 変更年月日 2000年 4月27日

[変更理由] 新規登録

住 所 神奈川県川崎市中原区新丸子915-15

氏 名 株式会社エイシーエス